Project Number: MIS - DMS - 0801

Dartmouth Harbormaster Database System

A Major Qualifying Project

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

_____

Millisent Fury Hopkins

_____

William D. Price

Date: January 12, 2009

_____

Professor Diane Strong, Project Advisor

This report represents the work of one or more WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement.
WPI routinely publishes these reports on its web site without editorial or peer review.

Table of Contents

Abstract 6

Executive Summary 7

Acknowledgments 9

Authorship Page 10

Chapter 1. Introduction 11

Chapter 2. Literature Review 13

2.1 Industry Overview 13

2.2 Technology Analysis 15

2.2.1 Database System Analysis 16

2.2.2 Front End System Analysis 17

2.2.3 Technology Analysis Conclusion 21

Chapter 3. As-is System 22

3.1 Overview of Current System 22

3.1.1 Business Processes 22

3.1.2 Quickbase Use Cases 22

3.1.3 Manual Records 38

3.2 Requirements Definition 39

3.3 Feasibility Analysis 40

3.3.2 Technical Feasibility 40

3.3.2 Economic Feasibility 41

3.3.2 Organizational Feasibility 42

3.4 Cost-Benefit Analysis 43

Chapter 4. Methods 44

4.1 SDLC 44

4.2 Work Schedule 45

4.3 Task List 46

Chapter 5. Design of the to-be System 47

5.1 Technology Decisions 47

5.2 Data Model 47

5.3 User Interface 50

Chapter 6. Documentation of the System 53

6.1 Using the System 53

6.1.1 Harbormaster System Dashboard 53

6.1.2 Listing Pages 54

6.1.3 Adding a person 57

6.1.4 Adding a Candidate to the Waitlist for Moorings 58

6.1.5 Adding a Boat 59

6.1.6 Adding a Mooring 60

6.1.7 Adding a New Dinghy 61

6.1.8 Adding a New Issue 62

6.1.9 Adding a New Inspection 63

6.1.10 Editing a mooring 64

6.1.11 Adding a New Ticket 65
6.1.12 Ticket Payment 66
6.1.13 Adding Rate Information 67
6.1.13 Listing Rate Information 67
6.1.14 Editing Rate Information 68
6.1.15 Generating a Bill 68
6.1.15 Editing Pages 69
6.1.16 Show Pages 75
6.2 Additional References 80

Chapter 7. Conclusions 81
7.1 Testing 81
7.2 Deployment 81
7.3 Future Recommendations 81
7.3 Group Reflection 82

Works Cited 86

Appendix I. Samples of Manual Documents 88
I.i Dinghy Renewal Form 89
I.ii Waterways Use Application: Form A (BOAT PERMIT) 90
I.iii Renewal Form 91

Appendix II. Technical Documentation 92

Appendix III. httpd-proxy_code 99

Appendix IV. User Manual 101

Appendix V. Meeting Agendas 122
V.i MQP Meeting Agenda September 11, 2008 123
V.ii MQP Meeting Agenda September 18, 2008 124
V.iii MQP Meeting Agenda October 2, 2008 125
V.iv MQP Meeting Agenda October 9, 2008 126
V.v MQP Meeting Agenda December 17, 2008 127

Appendix VI. Proposal Presentation 129

Appendix VII. Final Presentation 132

Appendix VIII. Letter of Acknowledgement 135

Table of Figures and Tables

Figure 1: System Request – Town of Dartmouth Harbormaster System 11
Figure 2: Add a new person 23
Figure 3: Add a person to waitlist 24
Figure 4: Add a new boat 25
Figure 5: Add a new mooring 26
Figure 6: Add a dinghy 27
Figure 7: Add a new issue 28
Figure 8: A customer requests a pump-out 29
Figure 9: Performing pump-outs and logging pump-out data 30
Figure 10: Generate the year-end pump-out report 31
Figure 11: Generate letters and invoices for renewals 32
Figure 12: Input inspection information for a mooring 33
Figure 13: Process a payment from a customer 34
Figure 14: Permittee requests change in mooring permit 35
Figure 15: Mooring permittee requests another be allowed to use their mooring 36
Figure 16: Add a new ticket 37
Figure 17: A customer pays a ticket 38
Figure 18: Gantt Chart 46
Figure 19: Entity Relationship Diagram – Dartmouth Harbormaster System 48
Figure 20: Add New Mooring Screenshot 51
Figure 21: Listing Boats Screenshot 51
Figure 22: New Boat Screenshot 52
Figure 23: Harbormaster System Dashboard 53
Figure 24: Listing Boats 54
Figure 25: Listing Moorings 54
Figure 26: Listing People 54
Figure 27: Listing Waitlists 55
Figure 28: Listing Tickets 55
Figure 29: Listing Inspections 55
Figure 30: Listing Dinghies 56
Figure 31: Listing Issues 56
Figure 32: Listing Bills 56
Figure 23: Adding a new person 57
Figure 24: Add a candidate to waitlist 58
Figure 25: Adding a new boat 59
Figure 26: Adding a new mooring 60
Figure 27: Adding a new dinghy 61
Figure 28: Adding a new issue 62
Figure 29: Adding a new inspection 63
Figure 30: Editing a mooring 64
Figure 31: Adding a new ticket 65
Figure 32: Ticket payment 66
Figure 34: Input Rate Information 67
Figure 34: Listing Rate Information 67

Figure 35: Editing Rate Information 68
Figure 33: Generate a bill 68
Figure 47: Edit Boat Page 70
Figure 48: Edit Person Page 71
Figure 49: Edit Waitlist Page 71
Figure 50: Edit Inspection Page 72
Figure 51: Edit Dinghy Page 73
Figure 52: Edit Issue Page 74
Figure 53: Edit Invoice Page 74
Figure 54: Show Boat Page 75
Figure 55: Show Mooring Page 76
Figure 56: Show Person Page 76
Figure 57: Show Waitlist Page 77
Figure 58: Show Ticket Page 77
Figure 59: Show Inspection Page 78
Figure 60: Show Dinghy Page 79
Figure 61: Show Issue Page 79
Figure 62: Show Invoice Page 80

Table 1: Cost-Benefit Analysis 43

Abstract

This project examined the database system in use at the Dartmouth Harbormaster's Office which is responsible for organizing diverse data for running a harbor of over 2,000 boats. The database system created in this project replaced a web-based application that did not meet the needs of the office with a free, mySQL based back-end and a web-based front-end. The final database system is fully-functional and running on the office's server for use by the Harbormaster and his assistants.

Executive Summary

The Town of Dartmouth Harbormaster office has spent years using its current system of tracking and manipulating data. These data include information regarding people, boats, moorings, and billing information. As an organization funded completely by the fees paid to them by boat owners who purchase waterways permits and pay for moorings, the Dartmouth Harbormaster must be vigilant to maintain accurate records that also allow efficient billing methods. While the current system serves this purpose, the employees of the Harbormaster office have realized that improvements could be made to both the efficiency of data tracking, and to operating costs.

The goal of this project was to replace this costly, non-expandable, web-based system with a free, mySQL based back-end and a web-based front-end. The first step towards reaching this goal was to meet with the Harbormaster and some of his assistants to research the use of the current system. By doing so the team was able to gain an understanding of the procedures in place and begin to build a list of requirements for the new system. By asking questions and exploring the as-is system the team was able to define use cases identifying the basic interactions with the system to aid in the design of the new one.

After observation and collection of requirements, the team met with the office staff to ensure the requirements gathered were accurate. The team then built an entity diagram to visually present the relations between data flowing through the system. From this data model, attributes for each entity were defined and the makings of a back-end were started.

After a successful proposal to the client, the project team began the development of the system outlined. A system was developed using Ruby on Rails, an open source web framework. The team split into two groups, one of which focused on the user interface and the other on the systems programming. Special consideration was given to the interface to emulate the legacy system and new features were developed by the systems programmer.

After successful unit testing, the team deployed the system to a web server located at the Dartmouth Harbormaster office. This web-based application can now be accessed via any computer with an Internet connection. Documentation was then gathered on the system and attached to this document for users and administrators of the new application.

Acknowledgments

Millisent Fury Hopkins and William Price would like to acknowledge the project sponsor, Town of Dartmouth Harbormaster, Steve Melo, and the project liaison, Julian Race for their continued trust and cooperation through the duration of this project. We would also like to thank our advisor, Diane Strong for her support and constructive feedback during our tight development schedule.

Authorship Page

Abstract – Millisent Fury Hopkins

Executive Summary – Millisent Fury Hopkins and William Price

Acknowledgments - Millisent Fury Hopkins

Introduction – Millisent Fury Hopkins and William Price

Industry Overview - Millisent Fury Hopkins

Technology Analysis - William Price

As-is System - William Price

Overview of Current System - Millisent Fury Hopkins

Feasibility & Cost-Benefit Analysis – William Price

Methods – Millisent Fury Hopkins

Design of the to-be System – Millisent Fury Hopkins and William Price

Documentation of the System – Millisent Fury Hopkins

Conclusions – William Price

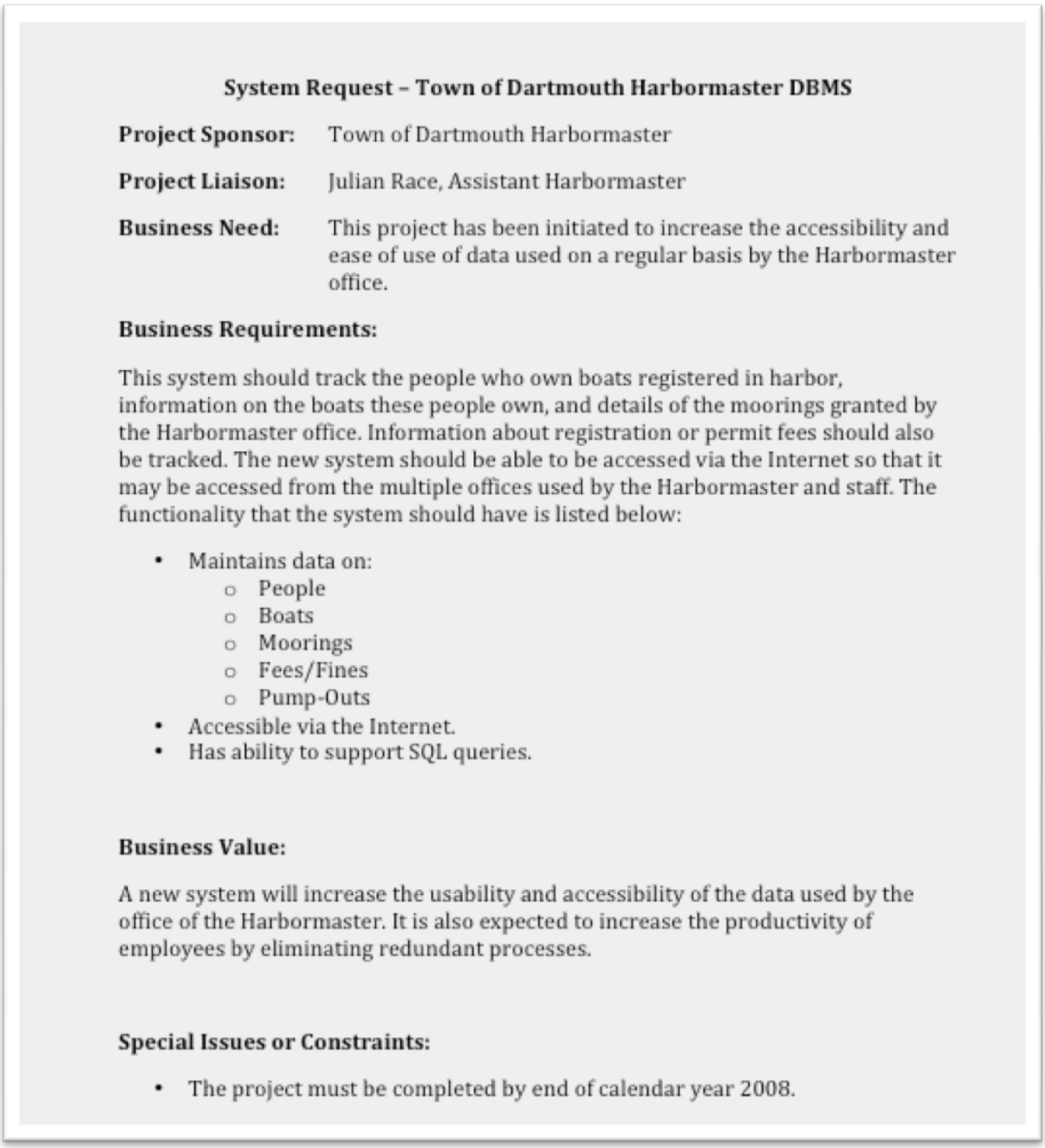
Presentations – William Price

Technical Documentation – William Price

User Manual - Millisent Fury Hopkins and William Price

Chapter 1. Introduction

The Harbormaster office issued a request for a system (Figure 1: System Request – Town of Dartmouth Harbormaster System) with some specific requirements for the system they desire. This request contains information on the project sponsor, the functionality they desire, and the timeframe for completing the project. This document is a response to this request and is broken into five main sections.



System Request – Town of Dartmouth Harbormaster DBMS

Project Sponsor: Town of Dartmouth Harbormaster

Project Liaison: Julian Race, Assistant Harbormaster

Business Need: This project has been initiated to increase the accessibility and ease of use of data used on a regular basis by the Harbormaster office.

Business Requirements:

This system should track the people who own boats registered in harbor, information on the boats these people own, and details of the moorings granted by the Harbormaster office. Information about registration or permit fees should also be tracked. The new system should be able to be accessed via the Internet so that it may be accessed from the multiple offices used by the Harbormaster and staff. The functionality that the system should have is listed below:

- Maintains data on:
  - People
  - Boats
  - Moorings
  - Fees/Fines
  - Pump-Outs
- Accessible via the Internet.
- Has ability to support SQL queries.

Business Value:

A new system will increase the usability and accessibility of the data used by the office of the Harbormaster. It is also expected to increase the productivity of employees by eliminating redundant processes.

Special Issues or Constraints:

- The project must be completed by end of calendar year 2008.

Figure 1: System Request – Town of Dartmouth Harbormaster System

After this Introduction is a Literature Review. In this chapter, our team has performed an Industry Review and a Technology Analysis. The Industry Review is a chance for the development team to demonstrate an understanding of the industry in which a new system would be used. The Technology Analysis is an overview of available technologies that could be used in the new system.

Following the Literature Review is an analysis of the as-is system. This analysis evaluates the data that is being tracked and assesses what is efficient and what could use improvement. This is achieved by examining the business processes performed, the steps performed in each process (called use cases), and a brief look at manual record keeping. This is followed by a definition of requirements for a new system, and an analysis of the risk involved in a project to develop a new system.

Chapter 4 is about the methods used to propose, design, and implement the new system. This chapter contains a project plan, a list of deliverables and deadlines, and an overview of the system development life cycle.

Chapter 5 presents the design of the new system, also known as the “to-be” system. It includes an outline of the data relationships and an overview of what the user interface of the new system looks like.

Chapter 6 provides screen-shots of the working system to provide a “feel” for the system developed. Along with the images are descriptions of functionality provided on that screen.

The new system has many advantages, which are discussed in the following sections.

Chapter 2. Literature Review

2.1 Industry Overview

The United States of America has trade and security interests in the surrounding waters. The shores of the United States are guarded on three different levels by federal, state and local agencies. There are 95,000 miles of coastline that require the attention of these agencies to promote transnational trade, protect the United States' borders and project military power.

On the federal level, the United States Coast Guard is an arm of the military that has continued to provide maritime defense since 1790 when its precursor, Revenue Marine was established by the First Congress of the United States (United States Coast Guard, 2008, p. 2). The Coast Guard has the following roles: "Maritime security, maritime safety, protection of natural resources, maritime mobility and national defense" (United States Coast Guard, 2008, p. 5). The Guard serves to protect and secure the nation's "ports, inland waterways, along the coast and any other maritime region where U.S. interests may be at risk" (United States Coast Guard, 2008, pp. 5-6). Its role in maritime security includes border control along the U.S. coast and enforcement of customs laws as well as all federal laws, treaties and any international agreements in the seas under U.S. jurisdiction.

Aside from border control and customs law enforcement, the U.S. Coast Guard is a protector of environmental resources in the seas surrounding the U.S. In 1976 the Magnuson-Stevens Fishery Conservation and Management Act was passed creating an Exclusive Economic Zone (EEZ). The EEZ is an increased area of 200 nautical miles of offshore natural ocean resources which need protection (United States Coast Guard, 2008, p. 8). Along with environmental protection, as National Recreational Boating Safety Coordinator, the Coast Guard plays a role in assuring the safety of boaters and their boats by boating regulations and education programs (United States Coast Guard, 2008, p. 9).

On the state level, the Massachusetts State Police Marine Section is responsible for patrolling the waters of the Charles and Mystic Rivers and the Harbor Islands National Park. In terms of law enforcement and protection, the Marine Section often provides escort to the U.S.S. Constitution, visiting cruise ships and Liquid Natural Gas tankers. The Marine Section also provides any presidential or diplomatic detail needed in the waters of the Boston Harbor, Martha's Vineyard and the Charles River (Commonwealth of Massachusetts, 2008).

If the need should arise, the Marine Section aids the Underwater Recovery Unit (URT) with search and rescue missions. The Massachusetts State Police Underwater Recovery Unit is a team of troopers specially trained in diving and other missions executed in the water. The divers of the URT are trained to dive in all weather conditions and water temperature in addition to small spaces such as drainage pipes. Decisions about whether or not to participate in a mission are left to the Team Commander as safety is the first priority of the URT. The URT contributes to missions inside and out of the Commonwealth for state agencies as well as federal offices like the Federal Bureau of Investigations, Secret Service and U.S. Customs (Commonwealth of Massachusetts, 2008).

The Massachusetts Environmental Police are in charge of coordinating programming to educate the public about boat safety. *Boat Massachusetts* is the manual written by the Environmental Police to provide a guide to the rules and waterways regulations for Massachusetts' boaters.

At the local level, the Harbormaster for Dartmouth is appointed by the mayor to supervise the harbor and 81 miles of coastline (Town of Dartmouth). The Dartmouth Harbormaster's office is overseen by the Dartmouth Waterways Commission along with the Dartmouth Waterways Department. The Harbormaster is also the Waterways Director and manages areas that touch the waterways, such as the various parks and beaches in Dartmouth (Melo, 2008). The purpose of these organizations is to create and enforce laws and regulations to maintain a safe and

enjoyable experience for those who use Dartmouth's waterways (Town of Dartmouth, 2007, p. 205).

Enforcement of the waterway and boating laws set forth by local, state and federal governments, including Massachusetts General Laws pertaining to harbors, docks, and waterways (M.G.L. Chapters 90B, 91, 102, 266) ensures the responsible use of the harbor (Town Of Dartmouth Waterways Management Department, 2005). The Harbormaster's office is staffed by the Harbormaster and nine assistant harbormasters who enforce these laws in conjunction with law enforcement agencies.

In addition to granting various permits to boats and their owners and enforcing mooring inspections, the office does its part to execute Dartmouth's "No Discharge" designation in the Harbor by pumping out vessels' septic systems. Thousands of gallons of sewage were kept out of the waterways in 2007 from participation in this project (Town of Dartmouth, 2007, p. 205). The Harbormaster's Office is funded through user-based fees, such as waterway user and mooring registration fees, and supplemental grants to assist with costs of waste removal (Town of Dartmouth, 2001, p. 165).

In the event of an emergency in the waters of Dartmouth Harbor and Buzzard's Bay, the staff of the Harbormaster's Office is trained in medical response. In support of Dartmouth Police and Fire Departments, United States Coast Guard and State Police the Harbormaster's Office will assist in medical response, mariner distress, emergency evacuation, and search and rescue (Melo, 2008).

2.2 Technology Analysis

This technology analysis section highlights the advantages and disadvantages of using different technologies in the new system. There are two main areas of technology we will be analyzing, the front-end and the back-end. These two systems work together to create an environment for the user to access and manipulate data (the back end) with an optimal user interface (the front end).

Below are some definitions of terms used in this section:

- Open Source: A piece of software where the source code (the code that makes up the software) is open to the public. These software packages are usually free of charge and have an Internet community of developers who add to the existing code to expand and add new items.
- Framework: Sets of existing code that can be used to make development easier for a specific purpose.
- SQL: Structured Query Language – A language used for database queries and data manipulation.

2.2.1 Database System Analysis

The back end of the system is the set of tables where the data are actually stored. The end user never actually interacts directly with the back end. All data are accessed and displayed by the front-end application. The back-end is like the storage warehouse where all the data are stored, and the front end is like an associate who goes and finds that data for you.

For the back-end analysis, we have made the assumption that the system must be inexpensive and have the capability to interact with a front-end web application.

2.2.1.1 MySQL

MySQL is an open source database that is used by many web developers and by many major corporations including Google, Yahoo, and YouTube (MySQL :: About MySQL, 2008). Because of its ease of use, many web frameworks have built-in adapters to connect easily to MySQL databases. MySQL is open source, but it is also offered by Sun Microsystems, which supports it and offers Enterprise options that a client can opt to purchase.

The basic mySQL server package, as well as some useful tools such as "MySQL Administrator" and "MySQL Query Browser" are available free of charge.

These free tools help not only with development, but even more so with the ongoing maintenance of the database.

MySQL Advantages:

- Well-documented and large support community.
- Proven excellence for supporting web development.
- Free database package and free administrator tools.
- Development team has experience.

MySQL Disadvantages:

- No graphic user interface for building tables.

2.2.1.2 Microsoft SQL Server Express

The Express edition of Microsoft SQL Server 2008 is a free SQL server that is offered for download on the Microsoft Developer Network. This software is developed and supported by the Microsoft Corporation. While the backing of a major corporation is generally a good thing, the Express edition of SQL Server 2008 is stripped of some functionality offered by other SQL Server versions (Anderson, 2008). The database is optimized for use with the .NET framework and has built in connectors for ASP web page design.

Microsoft SQL Server 2008 Advantages:

- Good starting point for a Microsoft platform.
- Interacts well with .NET applications.
- Free GUI tools to help manage database.

Microsoft SQL Server 2008 Disadvantages:

- 4GB data limit.
- Not intended to compete with full featured open source databases.

2.2.2 Front End System Analysis

The "front end" of a system is the part of a system with which the end user interacts directly. In the case of the system being designed currently, the front end is a web application that interacts with the database server to access the necessary

data. We are also assuming that the front-end application options should be relatively inexpensive, and must have web features.

2.2.2.1 PHP

PHP is short for the HTML-embedded scripting language PHP: Hypertext Preprocessor. An HTML-embedded language is a language where the scripts can be written and referenced directly in the HTML code that builds the web pages. This extremely popular language has grown to become one of the most commonly used languages for web applications. Originally developed in 1995 as a series of Perl scripts, PHP has gone through five release versions, the most recent, PHP 5.0 being released in July 2004 (The PHP Group, 2008). The language was influenced by C, Java, and Perl and has similar syntax.

PHP is used very commonly in applications that access databases. There is functionality for connecting to mySQL databases, as well as other types of back-end systems. Since it is a major open-source language, there is a large community of developers that help expand and support the language as well.

PHP Advantages:

- Uses syntax that is similar to many popular compiled languages.
- Can be embedded into HTML (useful because it cuts down on code files on the web server)

PHP Disadvantages:

- Development team would need to learn both PHP and HTML.
- Other options are more user friendly for database applications.

2.2.2.2 ASP .NET

ASP .NET is a language developed by the Microsoft Corporation that utilizes its .NET framework to create web pages. Many major companies utilize ASP to deliver their web pages, including myspace.com, Dell, monster.com and newegg.com (Microsoft, 2008). Microsoft provides a free development environment with

Microsoft Visual Web Developer that allows the developer to customize user interfaces more easily.

Since the Microsoft Corporation designed ASP, users can look forward to easier integration on Windows systems and support and documentation provided by a major corporation. ASP features flexibility in web design, and can be the framework for nearly any kind of website application, whether it requires a database or not.

ASP .NET Advantages:

- Has been in use for a while and has a lot of documentation and examples.
- Utilizes the .NET framework which the development team knows a little.
- Has a custom development environment.
- Integrates easily with other Microsoft systems.

ASP .NET Disadvantages:

- Works best with Microsoft SQL Server, which has a 4GB limit
- Harder to connect to open source databases.

2.2.2.3 Ruby on Rails

Ruby on Rails (RoR) is a web development framework launched first in December 2005 (Lenz, 2008). The framework is written in the scripting language Ruby that was developed in the early 1990's by Yukihiro Matsumoto (About Ruby, 2008). The Rails framework was built to help make the design of web-driven database applications easier and faster. This was accomplished by creating a framework that automatically generated many of the more tedious and boring sections of code. The goal is to allow the developer to focus more on the application functionality than on the long, but necessary code base.

Ruby on Rails is open source software that has a large community of involved developers. Over the years these developers helped to fix bugs and write new functionality into the language (Lenz, 2008). The Rails framework helps the developer by adhering to some programming principles that make the development of applications easier and more efficient. The most important of these principles is the "don't repeat yourself" principle, in which the program stores frequently used

code in one place and references it for multiple applications rather than forcing the copy and paste of code (Lenz, 2008).

Rails Advantages:

- Open source software community provides lots of support.
- Framework is designed for taking a database online.
- Built in support for many database packages (mySQL, Oracle, SQL Server)
- Allows focus on functionality instead of on configurations
- Built in relational integrity. Makes querying easier.

Rails Disadvantages:

- Development team would need to learn Ruby.

2.2.2.4 Django

Django, like Ruby on Rails, is a web development framework designed to help make the development of database driven web-applications easier. Unlike RoR however, this framework is written in the scripting language Python. Django was released as open source software in July 2005. The framework was developed in a news market, and thus had a great strength in delivering content based pages (Holovaty, 2007). This language has become another front-runner in the web development framework landscape, but it is sometimes overshadowed by the popularity of RoR.

Django Advantages:

- Template feature allows multiple pages to look the same with minimal effort.
- Django automatically connects to the database server and will send it the necessary commands to create the schema.

Django Disadvantages:

- The development team would need to learn.
- Depending on the requirements of the pages, there may be a need for an HTML template for each individual page on top of the application.

2.2.3 Technology Analysis Conclusion

With an understanding of the available options, technology decisions can be made that will produce the best product for the requirements. The requirements are outlined in Chapter 3, and our technology decisions to meet those requirements are in Chapter 5.

Chapter 3. As-is System

3.1 Overview of Current System

The Overview of the Current System is an assessment of what is currently in use at the Harbormaster Office. This assessment was completed in steps. First a list of business processes was made. These business processes were then each looked at in more detail to create Use Cases. These steps allow for a more accurate requirements definition.

3.1.1 Business Processes

A business process is defined as a group of tasks that fulfill the need of a business. When data need to be stored, or a bill needs to be generated, a business process is performed to fulfill that need. An understanding of business processes helps the development team to define requirements for a new system.

The following is a list of business processes observed:

- Enter New Person
- Add Candidate to Waitlist
- Add a New Boat
- Register a New Mooring
- Add new Dinghy
- Add New Issue
- Pump-out Request
- Processing Pump-out
- Generate Pump-out Report
- Generate Yearly Billing Process
- Add a New Inspection
- Process Bill Payment
- RFP: Change in Mooring Permit
- RFP: Mooring Use
- Ticket Payment

3.1.2 Quickbase Use Cases

The existing system stores electronic data in an online database called Quickbase that is offered by Intuit Inc. The Quickbase system is accessed using a webpage online, and has some relational integrity. Quickbase allows users to enter and manipulate data, however the user interface is cluttered, and the system lacks the flexibility desired.

Use cases are step-by-step walkthroughs of the major business processes performed in the Quickbase system. The use cases are used to analyze the strengths and weaknesses of the current system. Also use cases help the development team

understand the user interface, so a similar interface can be created for the new system.

3.1.2.1 Use Case #1

This use case outlines steps followed to add a new person to the system. Adding a new person happens when the Harbormaster's Office receives a candidate for the waitlist or a new boat owner requests a waterways use permit. It is also used when a current customer submits a request for another person to use their mooring, or when a boat and/or mooring is passed to a family member of a customer.

<table>
<tr><td colspan="2">Use Case Name: Enter New Person</td><td>ID: 1</td><td>Importance Level: High</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: Add a new person entry</td></tr>
<tr><td colspan="4">Trigger: A new customer submits a waitlist, mooring, or waterways permit application.</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td colspan="2">Major Outputs:
Description:</td><td>Destination:</td></tr>
<tr><td colspan="2">1. Application for:

-Waterways Use
-Waitlist
-RFP

-Dinghy</td><td>1. Customer</td><td colspan="2">1. New person entry</td><td>1. Person table</td></tr>
<tr><td colspan="6">Major Steps Performed:</td></tr>
<tr><td colspan="6">1. Open Quickbase
2. Select 'Add a New Person' from the Person drop-down menu.
3. Enter Name and Contact Information from application in associated fields
4. Enter in any Bills or Warnings
5. Click Save
6. File paper copy of application</td></tr>
</table>

Figure 2: Add a new person

3.1.2.2 Use Case #2

When a person is interested in having a mooring in Dartmouth Harbor and none are available, the steps in this use case are followed to add them to a waitlist in the system.

<table>
<tr><td colspan="2">Use Case Name: Add a Candidate to waitlist</td><td>ID: 2</td><td>Importance Level:
Medium</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: Adding a person to the Waitlist for moorings</td></tr>
<tr><td colspan="4">Trigger: Candidate submits waitlist application</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td>Major Outputs:
Description:</td><td>Destination:</td></tr>
<tr><td colspan="2">1. Form and payment</td><td>1. Candidate</td><td>1. New Person Entry

2. New Waitlist Entry</td><td>1. Person table
2. Waitlist table</td></tr>
<tr><td colspan="4">Major Steps Performed:</td></tr>
<tr><td colspan="4">1. Form is received from Candidate with payment of $15
2. Date and time received is filled out on form
3. Open Quickbase
4. Select 'Add a New Candidate' from the Waitlist dropdown menu
5. Click 'Browse choices...' for Person and select name if exists, otherwise click 'Add a New Person' (see Use Case #1)
6. Enter Desired Location and Vessel Information
7. Click 'Save' button
8. File paper copy of application</td></tr>
</table>

Figure 3: Add a person to waitlist

3.1.2.3 Use Case #3

This use case maps out the steps taken to add a boat to the system. Adding a boat occurs when a candidate applies for the waitlist for moorings, or when a current customer purchases a new boat. This use case is also used when a customer requests for another person to use his/her mooring, or when a new customer applies for a waterways use permit.

<table>
<tr><td colspan="2">Use Case Name: Add a New Boat</td><td>ID: 3</td><td>Importance Level:
High</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: Add a new boat</td></tr>
<tr><td colspan="4">Trigger: Customer submits Waterways Permit Application</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td colspan="2">Major Outputs:
Description:
Destination:</td></tr>
<tr><td colspan="2">1. Waterways Use App.</td><td>Customer</td><td>1. New Person Entry
-(if necessary)
2. New Boat Entry</td><td>1. Person Table

2. Boat Table</td></tr>
<tr><td colspan="4">Major Steps Performed:</td></tr>
<tr><td colspan="4">1. Open Quickbase
2. Click on 'Add a New Boat' from the Boats dropdown menu.
3. Enter Vessel Information
4. Select boat owner from dropdown menu, or create if doesn't exist (see Use Case #1)
5. Select existing mooring or add new one (if applicable [see Use Case #4])
6. Click Save.
7. File hard copy of application.</td></tr>
</table>

Figure 4: Add a new boat

3.1.2.4 Use Case #4

This use case describes the process for registering a new mooring when one becomes available for a customer on the waitlist.

Use Case Name: Register a New Mooring		**ID:** 4	**Importance Level:** High
Primary Actor: User			
Short Description: Add a new mooring			
Trigger: A mooring becomes available			
Type: EXTERNAL TEMPORAL			
Major Inputs: **Description:**	**Source:**	**Major Outputs:** **Description:**	**Destination:**
1. Mooring information 2. Candidate accepts mooring offer	1. user 2. Candidate	1. Mooring offer to candidate 2. New Mooring and Boat Registration Form	1. Candidate 2. Candidate
Major Steps Performed:			
1. Open Quickbase 2. Select 'Add a New Mooring' from Moorings dropdown menu 3. Input mooring number 4. Click 'Save' button 5. Click 'Add Inspection' button (see Use Case #11) 6. Click 'Browse choices...' for Owner of Record and select name 7. Select Status and Permit Category 8. Click 'Save' button			

Figure 5: Add a new mooring

3.1.2.5 Use Case #5

This use case explains the steps for adding a new dinghy to the system. Adding a new dinghy occurs when a customer submits an application for dinghy registration.

<table>
<tr><td colspan="2">Use Case Name: Add New Dinghy</td><td colspan="2">ID: 5</td><td>Importance Level: Medium</td></tr>
<tr><td colspan="5">Primary Actor: user</td></tr>
<tr><td colspan="5">Short Description: Add a dinghy</td></tr>
<tr><td colspan="5">Trigger: Customer submits Dinghy Registration Form</td></tr>
<tr><td colspan="5">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td colspan="2">Major Outputs:
Description:</td><td>Destination:</td></tr>
<tr><td colspan="2">1. Form and payment</td><td>1. Customer</td><td colspan="2">1. New person
(- if necessary)
2. New Dinghy</td><td>1. Person table

2. Dinghy table</td></tr>
<tr><td colspan="6">Major Steps Performed:</td></tr>
<tr><td colspan="6">1. Open Quickbase
2. Select 'Add a New Dinghy' from Dinghies dropdown menu
3. Click 'Browse choices...' for Owner and select name if exists, otherwise click 'Add a New Person' (see Use Case #1)
4. Select owner status
5. Input Dinghy Location and Information
6. Click 'Save' button
7. File paper copy</td></tr>
</table>

Figure 6: Add a dinghy

3.1.2.6 Use Case #6

When a customer has a complaint or a problem the following steps are taken to create an "issue" in the system.

<table>
<tr><td colspan="2">Use Case Name: Add New Issue</td><td>ID: 6</td><td>Importance Level: Medium</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: Add new a new issue</td></tr>
<tr><td colspan="4">Trigger: Customer calls, radios, or walks in with a problem.</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td colspan="2">Major Outputs:
Description: Destination:</td></tr>
<tr><td colspan="2">1. Complaint, issue, or Request</td><td>1. Customer</td><td>1. New Issue Record</td><td>1. Issues Table</td></tr>
<tr><td colspan="4">Major Steps Performed:</td></tr>
<tr><td colspan="4">1. Open Quickbase
2. Click on 'Add a New Issue' button on the Issues dropdown menu.
3. Enter in subject and issue description
4. Select boat, mooring, or person the issue relates to.
5. Click on Save.</td></tr>
</table>

Figure 7: Add a new issue

3.1.2.7 Use Case #7

The Dartmouth Harbor Office performs pump-outs of vessels' septic tanks in order to uphold the "No Discharge" law. A series of logs and forms are used for annual reporting to the town. This use case outlines the steps to log a pump-out request.

<table>
<tr><td colspan="2">Use Case Name: Pump-out Request</td><td>ID: 7</td><td>Importance Level: Medium</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: A customer requests a pump-out.</td></tr>
<tr><td colspan="4">Trigger: Customer calls, radios, or walks in and requests a pump-out.</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td colspan="2">Major Outputs:
Description: Destination:</td></tr>
<tr><td colspan="2">1. Pump-out Request
2. Vessel/Mooring Info</td><td>1. Customer
2. Customer</td><td>1. Updated Pump-out List</td><td>1. Office clipboard</td></tr>
<tr><td colspan="4">Major Steps Performed:</td></tr>
<tr><td colspan="4">1. Customer contacts the office and requests a pump-out, information is given.
2. Vessel name, location, and side of boat to be pumped are noted on the clipboard.
3. Customer has option of being added to weekly pump-out list.
4. Additional comments added (day of week request, etc.)
5. After pump-out performed (use case #8); the entry is crossed off the list.</td></tr>
</table>

Figure 8: A customer requests a pump-out

3.1.2.8 Use Case #8

To keep the waters of Dartmouth Harbor clean, the Harbormaster's Office offers the service of septic pump-outs. The following use case outlines the procedure of performing a pump-out.

<table>
<tr><td colspan="2">Use Case Name: Processing Pump Out</td><td>ID: 8</td><td>Importance Level: Medium</td></tr>
<tr><td colspan="4">Primary Actor: Assistant Harbormaster</td></tr>
<tr><td colspan="4">Short Description: Performing pump-outs and logging pump-out data</td></tr>
<tr><td colspan="4">Trigger: Daily Activity</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td colspan="2">Major Outputs:
Description:</td><td>Destination:</td></tr>
<tr><td colspan="2">1. Pump-out List</td><td>1. Office</td><td colspan="2">1. Updated Pump-out List</td><td>1. Pump-out Log book</td></tr>
<tr><td colspan="6">Major Steps Performed:</td></tr>
<tr><td colspan="6">1. Perform pump-out and log as complete
2. Repeat step 1 for rest of boats on Pump-out list
3. Return to Harbormaster's Office
4. Copy completed pump-outs from list to Pump-outs Log book</td></tr>
</table>

Figure 9: Performing pump-outs and logging pump-out data

3.1.2.9 Use Case #9

To continue receiving grant funding provided for waste removals, an end of the year report needs to be submitted with totals for gallons of waste removed by date. This use case describes the steps to generate that report.

<table>
<tr><td colspan="2">Use Case Name: Generate Pump-out Report</td><td>ID: 9</td><td>Importance Level:
Medium</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: Generate the year-end pump-out report for grant renewal</td></tr>
<tr><td colspan="4">Trigger: Yearly temporal report.</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description: Source:</td><td colspan="2">Major Outputs:
Description: Destination:</td></tr>
<tr><td>1. Pump-out log book</td><td>1. Office</td><td>1. Yearly Summary Report</td><td>1. Print</td></tr>
<tr><td colspan="4">Major Steps Performed:</td></tr>
<tr><td colspan="4">1. Open Microsoft Excel.
2. Manually enter data from pump-out log to Excel.
3. Repeat for each entry in the year of the report.
4. Print out report.
5. Click 'Save' button</td></tr>
</table>

Figure 10: Generate the year-end pump-out report

3.1.2.10 Use Case #10

As the Harbormaster's Office is funded through user fees, a yearly billing process needs to occur to receive payments. This use case explains the steps taken to generate yearly bills.

<table>
<tr><td colspan="2">Use Case Name: Generate Yearly Billing Process</td><td>ID: 10</td><td>Importance Level: High</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: Generate letters and invoices for all renewals for the coming year.</td></tr>
<tr><td colspan="4">Trigger: Every year mooring permitees are sent invoices for boats, moorings, dinghies, & wait.</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td colspan="2">Major Outputs:
Description: Destination:</td></tr>
<tr><td colspan="2">1. Person data
2. Boat data
3. Mooring data
4. Dinghy data
5. Waitlist data
6. Rates</td><td>1. Database
2. Database
3. Database
4. Database
5. Database
6. Harbor Office</td><td>1. Invoice</td><td>1. Customer</td></tr>
<tr><td colspan="4">Major Steps Performed:</td></tr>
<tr><td colspan="4">1. Generate CSV files of Quickbase tables.
2. Upload Person, Boat, and Mooring CSV files to MS Access and create relationships.
3. Use MailMerge feature to generate a .doc file of invoices.
4. Manually scan each invoice for errors.
5. Update .doc file with any updates or error corrections.
6. Upload Dinghy CSV to MS Access.
7. Use MailMerge feature to generate .doc file of Dinghy invoices.
8. Upload Waitlist CSV to MS Access.
8. Use MailMerge feature to generate .doc file of Waitlist invoices.
9. Print out the three .doc files.
10. Manually sort the papers to put the invoices for the same person together.
11. Stuff envelopes, manually stamp, and then mail.</td></tr>
</table>

Figure 11: Generate letters and invoices for renewals

3.1.2.11 Use Case #11

To ensure the safety of those who use Dartmouth Harbor, inspections of moorings are required every three years by law. The following use case goes through the process of adding an inspection to a moorings record in the system.

<table>
<tr><td colspan="2">Use Case Name: Add an Inspection</td><td>ID: 11</td><td>Importance Level: Low</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: Input inspection information for a mooring</td></tr>
<tr><td colspan="4">Trigger: New inspection recently completed</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description: Source:</td><td colspan="2">Major Outputs:
Description: Destination:</td></tr>
<tr><td>1. Inspection report</td><td>1. Inspection company</td><td>1. Inspection information</td><td>1. Inspection table</td></tr>
<tr><td colspan="4">Major Steps Performed:</td></tr>
<tr><td colspan="4">1. Open Quickbase
2. Select 'Add a New Inspection' from Inspections drop down
3. Click 'Browse choices...' for Related Mooring and select number
4. Input inspection date and Inspector and related mooring information
5. Click 'Save' button</td></tr>
</table>

Figure 12: Input inspection information for a mooring

3.1.2.12 Use Case #12

Once bills have been sent to customers, their payments need to be recorded in the system. This use case explains the steps for processing a bill payment.

<table>
<tr><td colspan="2">Use Case Name: Process Bill Payment</td><td>ID: 12</td><td>Importance Level: High</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: Process a payment from a customer.</td></tr>
<tr><td colspan="4">Trigger: Receive check in mail from customer.</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td colspan="2">Major Outputs:
Description: Destination:</td></tr>
<tr><td colspan="2">1. Invoice
2. Payment</td><td>1. Customer
2. Customer</td><td>1. Updated Balance</td><td>1. Database</td></tr>
<tr><td colspan="4">Major Steps Performed:
1. Using information on invoice, query database for person.
2. Enter in the amount paid in the corresponding columns.
-May need to use multiple forms, since dinghies are unrelated.
3. Save updates in database.
4. Mail the check to the town Collector.</td></tr>
</table>

Figure 13: Process a payment from a customer

3.1.2.13 Use Case #13

This use case describes the steps for changing a mooring permit in the system. This would happen in the event of a transfer to a family member, or the purchase of a new boat.

<table>
<tr><td colspan="2">Use Case Name: RFP: Change in Mooring Permit</td><td>ID: 13</td><td>Importance Level: Medium</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: A user requests a change in their mooring permit.</td></tr>
<tr><td colspan="4">Trigger: RFP form received in mail.</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description: Source:</td><td colspan="2">Major Outputs:
Description: Destination:</td></tr>
<tr><td>1. RFP: Change in Permit</td><td>1. Customer</td><td>1. Updated Mooring Permit
2. New Boat Entry
-If necessary
3. New Person Entry
-If necessary</td><td>1. Mooring Table
2. Boat Table

3. Person Table</td></tr>
<tr><td colspan="4">Major Steps Performed:</td></tr>
<tr><td colspan="4">1. Determine type of RFP:
- Transfer to Family Member
- Approve new vessel of record for mooring.
- Other
2. Alter Quickbase entry on mooring to reflect change.
- Change person linked to mooring in case of transfer
- Change in boat linked to mooring in case of new vessel of record.
3. Create new person or boat if necessary (see use case #1 and #3 respectively.)</td></tr>
</table>

Figure 14: Permittee requests change in mooring permit

3.1.2.14 Use Case #14

In the event a customer would like to lend his/her mooring to a friend for part of a season, the following steps are taken to input that information in the system.

<table>
<tr><td colspan="2">Use Case Name: RFP: Mooring Use</td><td>ID: 14</td><td>Importance Level: Low</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: Mooring permittee requests another be able to use his/her mooring</td></tr>
<tr><td colspan="4">Trigger: RFP received from mooring permittee</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description: Source:</td><td colspan="2">Major Outputs:
Description: Destination:</td></tr>
<tr><td>1. RFP: Mooring Use form</td><td>1. Mooring Permittee</td><td>1. Permitted User information</td><td>1. Person table</td></tr>
<tr><td colspan="4">Major Steps Performed:
1. Open Quickbase
2. Select 'Find People' from People dropdown
3. Select the Permitted User's name if exists, otherwise click 'Add a New Person' (see Use Case #1)
4. Set 'Status', 'rfp status' and 'RFP Type'
5. Click 'Save' button
6. Select 'Add a New Boat' from Boats dropdown, (see Use Case #3)
7. Input Permittee's mooring number in location field
8. Click 'Save' button</td></tr>
</table>

Figure 15: Mooring permittee requests another be allowed to use their mooring

3.1.2.15 Use Case #15

To ensure the safety of those in the waters of Dartmouth Harbor, the Harbormaster's Office is also in charge of handing out violations to those not following the law. This use case outlines the process for entering a ticket into the system.

<table>
<tr><td colspan="2">Use Case Name: Add a New Ticket</td><td>ID: 15</td><td>Importance Level:
Medium</td></tr>
<tr><td colspan="4">Primary Actor: User</td></tr>
<tr><td colspan="4">Short Description: Create a new ticket entry</td></tr>
<tr><td colspan="4">Trigger: Person commits an offense and is given a citation.</td></tr>
<tr><td colspan="4">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td colspan="2">Major Outputs:
Description:
Destination:</td></tr>
<tr><td colspan="2">1. Ticket</td><td>1. HM Employee</td><td>1. New ticket entry</td><td>1. Ticket table</td></tr>
<tr><td colspan="4">Major Steps Performed:</td></tr>
<tr><td colspan="4">1. Open Quickbase
2. Select 'Add a New Ticket' from the Ticket drop-down menu.
3. Enter Ticket Number and select an existing person (or create new if necessary)
4. Enter in amount of citation
5. Click Save
1. Open Quickbase
2. Select 'Add a New Ticket' from the Ticket drop-down menu.
3. Enter Ticket Number and select an existing person (or create new if necessary)</td></tr>
</table>

Figure 16: Add a new ticket

3.1.2.16 Use Case #16

When a person commits an offense in the waters of Dartmouth Harbor a citation/ticket is issued. A person can either mail in a check to pay the citation or appear in person. The following outlines the steps for processing a ticket payment in the system.

<table>
<tr><td colspan="2">Use Case Name: Ticket Payment</td><td colspan="2">ID: 15</td><td>Importance Level: Low</td></tr>
<tr><td colspan="5">Primary Actor: User</td></tr>
<tr><td colspan="5">Short Description: A customer pays a ticket or citation.</td></tr>
<tr><td colspan="5">Trigger: A customer makes a payment for a fine.</td></tr>
<tr><td colspan="5">Type: EXTERNAL TEMPORAL</td></tr>
<tr><td colspan="2">Major Inputs:
Description:</td><td>Source:</td><td colspan="2">Major Outputs:
Description:</td><td>Destination:</td></tr>
<tr><td colspan="2">1. Ticket
2. Amount Paid</td><td>1. Customer
2. Customer</td><td colspan="2">1. Updated Ticket Entry</td><td>1. Ticket Table</td></tr>
<tr><td colspan="6">Major Steps Performed:</td></tr>
<tr><td colspan="6">1. Open Quickbase.
2. Select 'Find a Ticket' from the Tickets dropdown menu.
3. Enter in the ticket number and bring up the entry.
4. Click on the 'Edit' button to make fields editable.
5. Enter the amount paid and the date paid in the respective fields.
6. Click Save.</td></tr>
</table>

Figure 17: A customer pays a ticket

3.1.3 Manual Records

Although the Harbormaster's Office currently employs an online system to keep its data, there is a list of paper documents that circulates through the office. The information on these documents is usually input into the system and a hard-copy filed away in a cabinet or binder for record purposes. Even though the development team would like to eliminate the paperwork flowing through the office, it does not appear possible at this time. On the other hand, by providing a system that is more intuitive and easier to use, there is an expectation that paper forms will not need to be the in-between step anymore and information can be put

directly into the system. Copies of some of said documents are provided in the appendix to this proposal.

3.2 Requirements Definition

The following requirements list the features needed in the system. They are broken into two sections. The Core Requirements are the requirements the system must have in order to function. The Expanded Requirements are requirements that will be included if feasible within the timeframe.

Core Requirements:

1. The system must store name and contact information about the customers of the Harbormaster's Office.
2. The system must store information about boats.
3. The system must store information about moorings, including number, owner and inspection date.
4. The system must store information on dinghies such as contact information of owner, and information and location of dinghy.
5. The system must store ticket/violation data.
6. The system must produce reports about people, moorings and boats.
7. These reports will draw information from other tables such as inspections, tickets, and warnings.
8. The system must support multiple users with log-ins.
9. The system must be secured with appropriate passwords and administrative restrictions.
10. The system must have a web-based front-end and be a SQL-based relational database management system.
11. The system must be accessible from any computer with an Internet connection.

Expanded Requirements (In order of priority):

12. The system must support mail-merge activities for sending out registration renewals and bills.
13. The system must store photographs of boats.
14. The system must provide a way to track/store transient vessel information.

3.3 Feasibility Analysis

We performed a feasibility analysis for the Dartmouth Harbormaster System project to evaluate risk. There are three main areas covered in this analysis: technical, economic, and organizational feasibility. No project comes without inherent risk; the goal of this report is help understand where the risks arise and the possible repercussions.

3.3.2 Technical Feasibility

Technically, the development of the Dartmouth Harbormaster System is feasible; however there are a reasonable amount of risks involved.

Dartmouth Harbormaster's risk regarding computerized databases is low.

- The Harbormaster employees have experience using a web-based database.

Dartmouth Harbormaster's risk regarding familiarity with the technology is high.

- The development team will be using web application technology that is new to them.
- The development team does have experience using the database server.
- Few Harbormaster employees have technical experience with database or web development.
- Development tools and references are available regarding all software to be used in the system.

The project size is considered medium risk.

- The development team will have two members.
- Business user involvement will be required.
- The project timeframe is set for completion by the end of the calendar year, allowing approximately 2.5 months for development and testing.

The risk regarding compatibility to existing systems is medium.

- The new system will not be interacting with any existing software since it is being used as a replacement.
- Open standards will allow future compatibility and integration into new systems the Harbormaster's office may want to incorporate into its business.
- The ability to port old data to the new system is undetermined.

3.3.2 Economic Feasibility

The best way of estimating the economic feasibility of a project is to perform a cost-benefit analysis, which applies monetary values to the costs of having the project done and the ongoing maintenance. It also applies monetary values to the benefits of the new system including the savings gained by increased productivity. A cost benefit analysis usually looks ahead between 3 and 10 years at the economic impact of a project. In this case, a 5-year analysis was conducted.

Please see Table 1 for the cost benefit analysis of the Dartmouth Harbormaster System project. A conservative approach was taken for the calculations, meaning the numbers used were the low end of estimates for monetary values. For example, if the new system would save 1-3 man-hours per day, calculations were computed using the numbers for 1 man-hour.

Some highlights of the cost-benefit analysis are:

- Return on investment over 5 years is 30.95%.
- Total benefit after five years is $21,237.80.
- Break-even occurs after 3.49 years.

These numbers indicate that this system will pay for itself through savings and increased productivity in three and a half years. The system will also grant over a 30% return on the initial investment. The development of this system is economically feasible.

3.3.2 Organizational Feasibility

This project yields a medium risk from the organizational perspective. The objective of the system is in line with the mission of the Harbormaster and will certainly help attain the goals set forth by the Harbormaster office. Time saved by using the new system will allow for more projects on the water and more services provided to clients.

The project champion, Assistant Harbormaster Julian Race, is very well equipped to act as the project liaison and will be qualified to use and maintain the system in the future after the system goes live. Mr. Race has gained the support of not only the Harbormaster's office but also of the town of Dartmouth as well.

The users of this system, the Harbormaster employees, will be well equipped to use the new system. There is question, however, as to how well it will be received. The methods used in this office may be inefficient, but most of the staff have accepted them and grown used to the way things are done. Adoption of the new system may not be without resistance or confusion among the users.

3.4 Cost-Benefit Analysis[1]

	2008	2009	2010	2011	2012	Total
Benefits						
Quickbase Savings	0.00	1,594.65	1,594.65	1,594.65	1,594.65	6,378.60
Savings from increased employee efficiency	0.00	3,674.80	3,674.80	3,674.80	3,674.80	14,699.20
Reduced Supply Costs	0.00	40.00	40.00	40.00	40.00	160.00
Total Benefits	**0.00**	**5,309.45**	**5,309.45**	**5,309.45**	**5,309.45**	**21,237.80**
Development Costs						
Development Labor	5,000.00	0.00	0.00	0.00	0.00	5,000.00
Cost of Time Spent w/ Developers	237.89	0.00	0.00	0.00	0.00	237.89
Server Cost	2,000.00	0.00	0.00	0.00	0.00	2,000.00
Total Development Costs	**7,237.89**	**0.00**	**0.00**	**0.00**	**0.00**	**7,237.89**
Operational Costs						
Server Maintenance	0.00	312.50	312.50	312.50	312.50	1,250.00
Operational Labor	0.00	1,459.12	1,459.12	1,459.12	1,459.12	5,836.48
Training Costs	0.00	1,122.40	350.75	210.45	210.45	1,894.05
Total Operational Costs	**0.00**	**2,894.02**	**2,122.37**	**1,982.07**	**1,982.07**	**8,980.53**
Total Costs	**7,237.89**	**2,894.02**	**2,122.37**	**1,982.07**	**1,982.07**	**16,218.42**
Total Benefits - Total Cost	**(7,237.89)**	**2,415.43**	**3,187.08**	**3,327.38**	**3,327.38**	**5,019.38**
Cumulative Net Cash Flow	**(7,237.89)**	**(4,822.46)**	**(1,635.38)**	**1,692.00**	**5,019.38**	
Reurn on Investment (ROI	**30.95%**					
Break-Even Point	**3.49 years**					

Table 1: Cost-Benefit Analysis

[1] Explanation of values:

- Savings from increased efficiency: Calculated at one-man hour ($14.03) saved per day.
- Cost of time spent with Development team: Calculated at 2 Assistant Harbormaster (Asst. HM) and 1 HM hour per visit for 11 visits (14.03+14.03+31.41)*11 visits.
- Operational Labor: Estimated 2 man-hours of Asst. HM work per work.
- Training cost: Budgeted 5 hours of training time per employee in year #2. Year #3 allows for 20 hours of training. Years #4 and #5 allow for 15 hours. Also includes 30 hours for the system admin to learn Ruby on Rails.

Chapter 4. Methods

4.1 SDLC

The primary method used to guide this project is the System Development Life Cycle (SDLC), which is a process for developing systems. SDLC is a standard set of steps to follow for systems development, specifically planning, analysis, design and implementation.

In the planning phase, the development team gathers requirements for the system, in other words, what the system needs to do or what functionality it needs to contain. In addition to requirements, the team gets a sense of the value of the system for the business and sets out a plan for the project. The analysis phase is where the team investigates the costs and benefits of implementing a new system (Dennis, Wixom, & Roth, 2006).

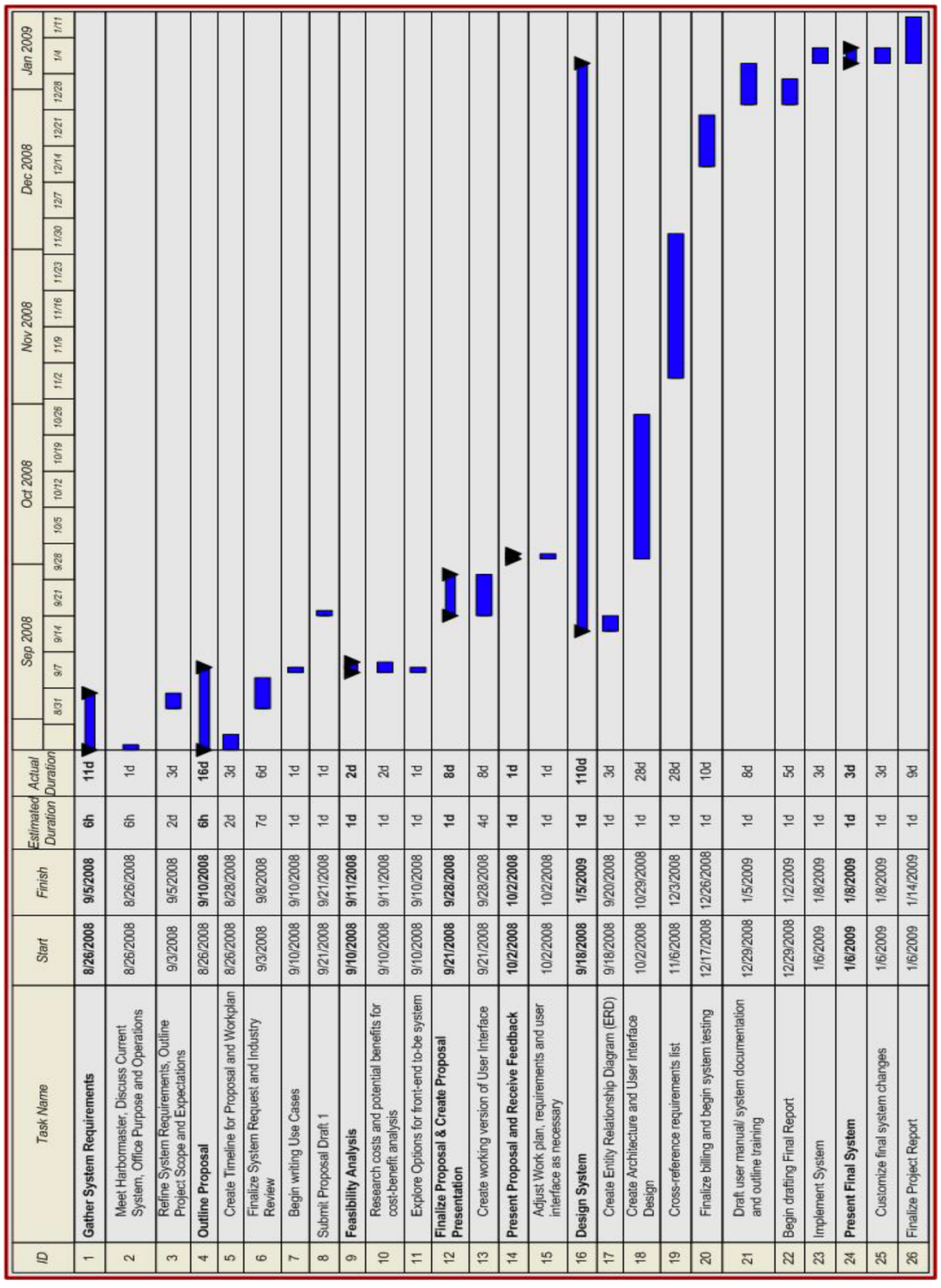
In the design phase the team does just that, designs the system in the physical and technical sense by creating the architecture and user interface designs. The design phase also includes any testing and fine tuning to specifications. The final phase of the SDLC is the implementation phase, the time when the system is put in place and training for it begins (Dennis, Wixom, & Roth, 2006).

4.2 Work Schedule

Milestone/Deliverable	Target Date
1. Meet with Harbormaster to discuss system needs	August 26, 2008
2. Refine system requirements and project scope	September 3, 2008
3. Complete System Request and Industry Review	September 4, 2008
4. Complete Use Case, Feasibility and Technology Analyses	September 10, 2008
5. Create Data Model	September 14, 2008
6. Draft User Interface	September 26, 2008
7. Present Proposal	October 1, 2008
8. Begin System Architecture design	October 2, 2008
9. Begin User Interface design	November 5, 2008
10. Begin System Testing	December 3, 2008
11. Implement System	December 29, 2008
12. Final Presentation	January 6, 2009
13. Finalize Project Report	January 12, 2009

4.3 Task List

The following figure is a schedule of tasks for project completion. The schedule runs Sunday to Saturday from August 26, 2008 to January 14, 2009.



ID	Task Name	Start	Finish	Estimated Duration	Actual Duration
1	**Gather System Requirements**	**8/26/2008**	**9/5/2008**	**6h**	**11d**
2	Meet Harbormaster, Discuss Current System, Office Purpose and Operations	8/26/2008	8/26/2008	6h	1d
3	Refine System Requirements, Outline Project Scope and Expectations	9/3/2008	9/5/2008	2d	3d
4	**Outline Proposal**	8/26/2008	**9/10/2008**	**6h**	**16d**
5	Create Timeline for Proposal and Workplan	8/26/2008	8/28/2008	2d	3d
6	Finalize System Request and Industry Review	9/3/2008	9/8/2008	7d	6d
7	Begin writing Use Cases	9/10/2008	9/10/2008	1d	1d
8	Submit Proposal Draft 1	9/21/2008	9/21/2008	1d	1d
9	**Feasibility Analysis**	**9/10/2008**	**9/11/2008**	**1d**	**2d**
10	Research costs and potential benefits for cost-benefit analysis	9/10/2008	9/11/2008	1d	2d
11	Explore Options for front-end to-be system	9/10/2008	9/10/2008	1d	1d
12	**Finalize Proposal & Create Proposal Presentation**	**9/21/2008**	**9/28/2008**	**1d**	**8d**
13	Create working version of User Interface	9/21/2008	9/28/2008	4d	8d
14	**Present Proposal and Receive Feedback**	**10/2/2008**	**10/2/2008**	**1d**	**1d**
15	Adjust Work plan, requirements and user interface as necessary	10/2/2008	10/2/2008	1d	1d
16	**Design System**	**9/18/2008**	**1/5/2009**	**1d**	**110d**
17	Create Entity Relationship Diagram (ERD)	9/18/2008	9/20/2008	1d	3d
18	Create Architecture and User Interface Design	10/2/2008	10/29/2008	1d	28d
19	Cross-reference requirements list	11/6/2008	12/3/2008	1d	28d
20	Finalize billing and begin system testing	12/17/2008	12/26/2008	1d	10d
21	Draft user manual/ system documentation and outline training	12/29/2008	1/5/2009	1d	8d
22	Begin drafting Final Report	12/29/2008	1/2/2009	1d	5d
23	Implement System	1/6/2009	1/8/2009	1d	3d
24	**Present Final System**	**1/6/2009**	**1/8/2009**	**1d**	**3d**
25	Customize final system changes	1/6/2009	1/8/2009	1d	3d
26	Finalize Project Report	1/6/2009	1/14/2009	1d	9d

Figure 18: Gantt Chart

Chapter 5. Design of the to-be System

5.1 Technology Decisions

The new system is built using mySQL for back-end. This is a free and dynamic database server that has the web capabilities that are required. The front-end of the system is written using Ruby on Rails (RoR). This front-end decision allows for future expandability and contains features that will assist in web development.

5.2 Data Model

A data model is a diagram that shows the different types of data to be stored (entities) and their relationships to each other. Each square represents an entity, with the name being in the center. A line represents a relationship to another entity. The values reading “1” and “M” denote the kind of relationship the line represents. For example, if one person can own multiple dinghies, there is a “1” next to person, and an “M” next to Dinghy.

This model is used to create the back-end of the system. The relationships indicate how data is pulled from multiple tables.

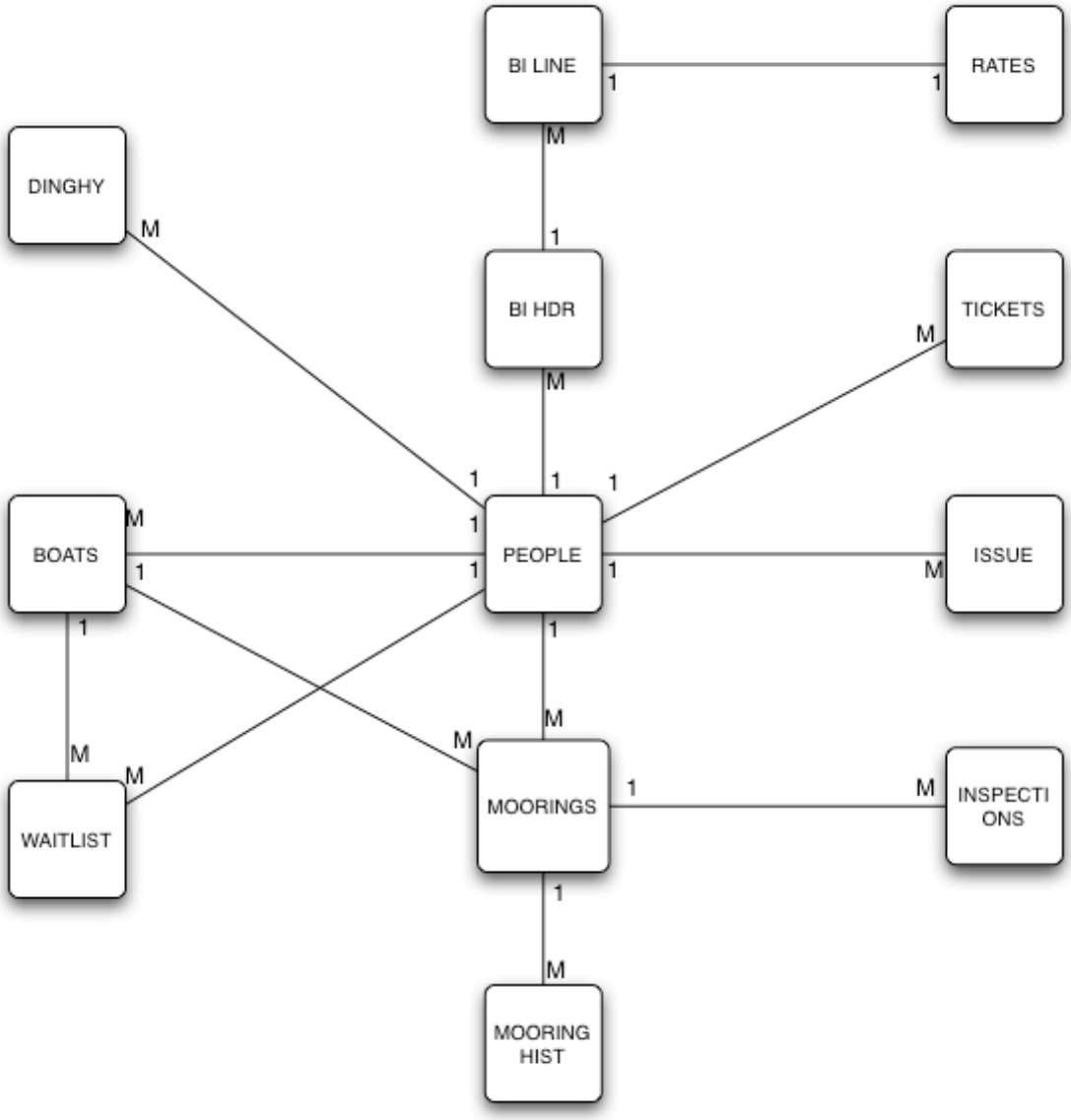


Figure 20: Entity Relationship Diagram – Dartmouth Harbormaster System

The following is a list of field names for the to-be system.

Person

- Customer ID (PK)
- First Name
- Last Name
- Suffix
- Street
- City
- State
- Zip Code
- Phone
- Alt Phone
- E-Mail
- Resident Status
- Alt address
- Notes

Boat

- Boat ID (PK)
- Vessel Name
- Owner (FK)
- Length
- Draft
- Color
- Type
- Vessel Year
- Manufacturer
- Registration
- Pump-out
- Location
- Active
- Notes
- Beam

Mooring

- Mooring Number (PK)
- Owner (FK)
- Site Depth
- Latitude
- Longitude
- Permit type
- Notes

Tickets

- Ticket Number (PK)
- Violator (FK)
- Date
- Time
- Boat ID (FK)
- Location
- Officer
- Violation Code
- Fine Total
- Notes

Mooring Inspection

- Inspection ID (PK)
- Mooring Number (FK)
- Inspection Date
- Inspector
- Weight
- Type
- Bottom Chain Size
- Bottom Chain Length
- Bottom Chain Condition
- Top Chain Size
- Top Chain Length
- Top Chain Condition
- Pennant Size
- Pennant Length
- Pennant Condition
- Mooring Eye Condition
- Notes

Dinghy

- Dinghy ID (PK)
- Owner (FK)
- Status
- Location
- Permit Type
- Decal Number
- Length
- Color
- Manufacturer
- Model
- HIN
- Type
- Registration
- Reg. State
- Features/Identifying Marks
- Notes

Waitlist

- Waitlist Number (PK)

- Person (FK)
- Boat (FK)
- Entry time
- Status
- Desired Location
- Notes
- Assigned location
- Assigned date

Issues

- Issue ID (PK)
- Person (FK)
- Status
- Due Date
- Assigned To
- Priority
- Subject
- Description
- Work log

BI HDR (Billing Header)

- Invoice # (PK)
- Person (FK)
- Date
- Amount Paid

BI LINE (Billing Line Detail)

- Billing Line # (PK)
- Invoice (FK)
- Rate Type (FK)
- Amount
- Description

RATES

- Rate Year (PK)
- Rate Type (PK)
- Rate Amount

MOORING HIST

- Mooring Hist # (PK)
- Mooring (FK)
- Owner (PK)
- Boat (FK)

5.3 User Interface

The user interface is the part of the application that appears in the web browser. One of the major user interface goals is to utilize a similar style to Quickbase to facilitate user learning of the new system. In order to assure this the user interface for the system was designed by following the layout of information in Quickbase's pages. The following figures are examples of what some pages look like in the final system.

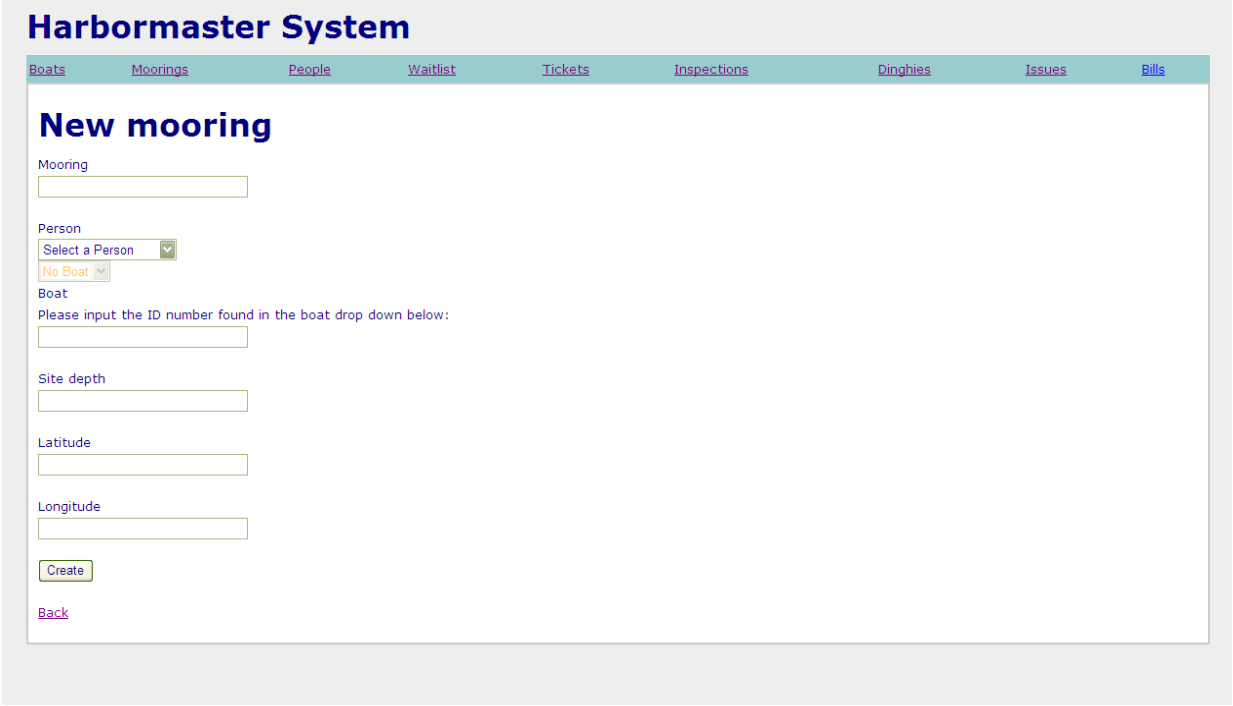


Figure 21: Add New Mooring Screenshot

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Search By: VESSEL_NAME
Listing boats
Year Name Type Length Color Owner Name Mooring Number
2008 Guiding Light POWER 30.0 White Price, William 1A-1A Show | Edit
2008 Waiting Boat POWER 15.0 White Price, William --- Show | Edit
New boat

Figure 22: Listing Boats Screenshot

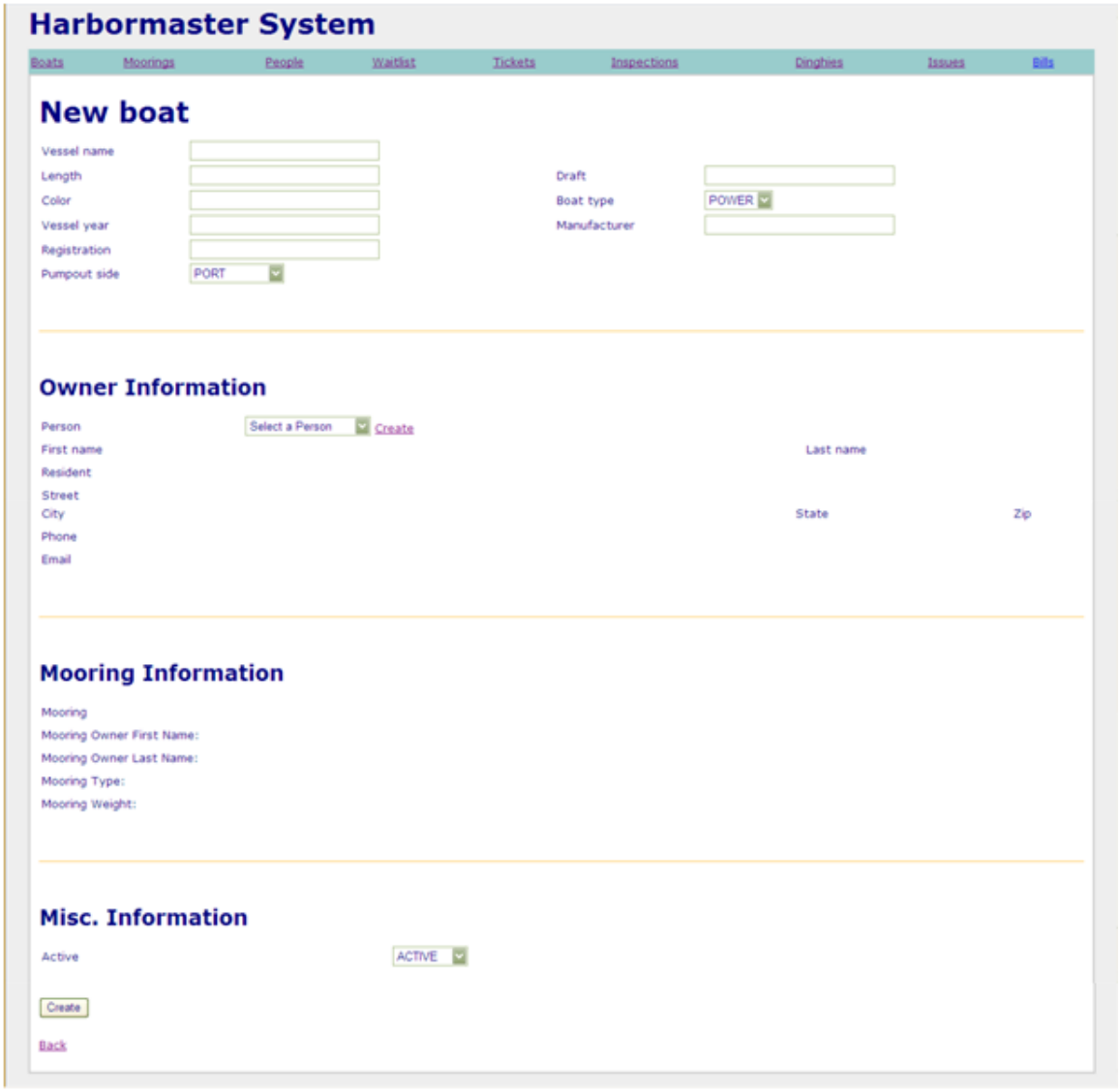


Figure 23: New Boat Screenshot

Chapter 6. Documentation of the System

6.1 Using the System

The following are screenshots of the completed Dartmouth Harbormaster system. The images listed map to many of the use cases in section 3.1.2 Quickbase Use Cases.

6.1.1 Harbormaster System Dashboard

To provide familiarity for the Dartmouth Harbormaster and his assistants, a dashboard was provided for the homepage of the system similar to the dashboard in Quickbase. This dashboard lists expiring inspections and open issues in the system providing the user with pending items requiring attention.

Figure 24: Harbormaster System Dashboard

6.1.2 Listing Pages

In addition to the dashboard page, listing pages were also included to mimic some of the basics of the Quickbase system. These listing pages are index pages for the various data categories in the system. The following are screenshots of the listing pages for Boats, Moorings, People, Waitlist, Tickets, Inspections, Dinghies, Issues and Bills.

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Search By: VESSEL_NAME
Listing boats
Year Name Type Length Color Owner Name Mooring Number
2008 Guiding Light POWER 30.0 White Price, William 1A-1A Show | Edit
2008 Waiting Boat POWER 15.0 White Price, William --- Show | Edit
New boat

Figure 25: Listing Boats

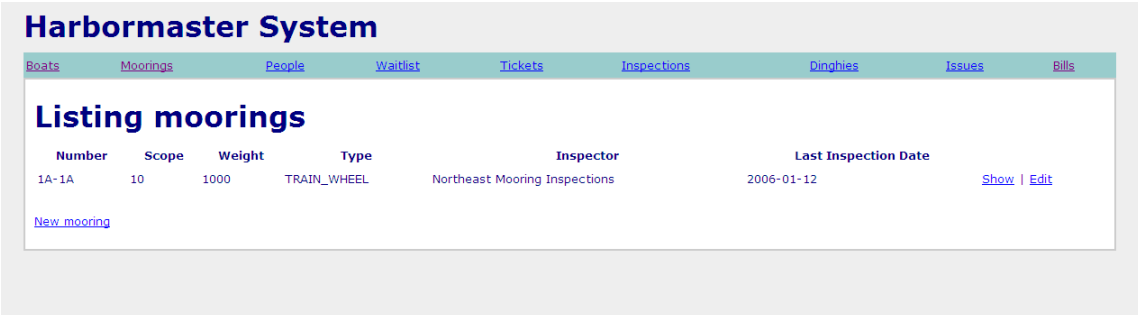

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Listing moorings
Number Scope Weight Type Inspector Last Inspection Date
1A-1A 10 1000 TRAIN_WHEEL Northeast Mooring Inspections 2006-01-12 Show | Edit
New mooring

Figure 26: Listing Moorings

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Search By: LAST_NAME
Listing people
Name Phone number Email City State
Price, William N/A wprice@wpi.edu Worcester MA Show|Edit
Race, Julian 5082643812 julianrace@gmail.com South Dartmouth MA Show|Edit
New person

Figure 27: Listing People

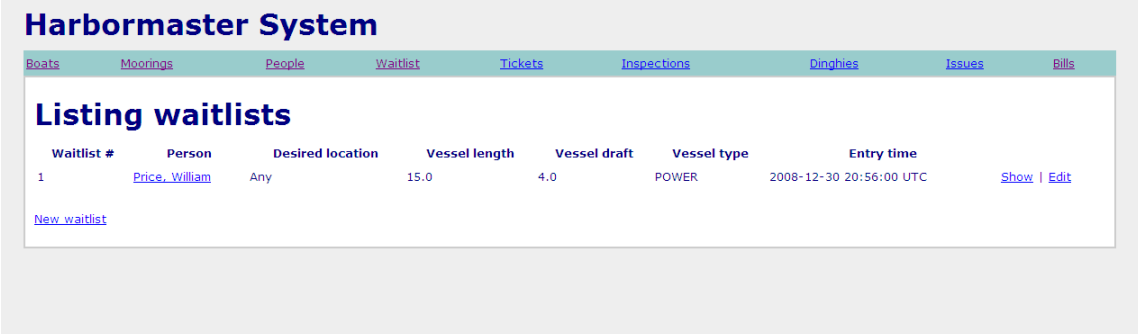

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Listing waitlists
Waitlist # Person Desired location Vessel length Vessel draft Vessel type Entry time
1 Price, William Any 15.0 4.0 POWER 2008-12-30 20:56:00 UTC Show | Edit
New waitlist

Figure 28: Listing Waitlists

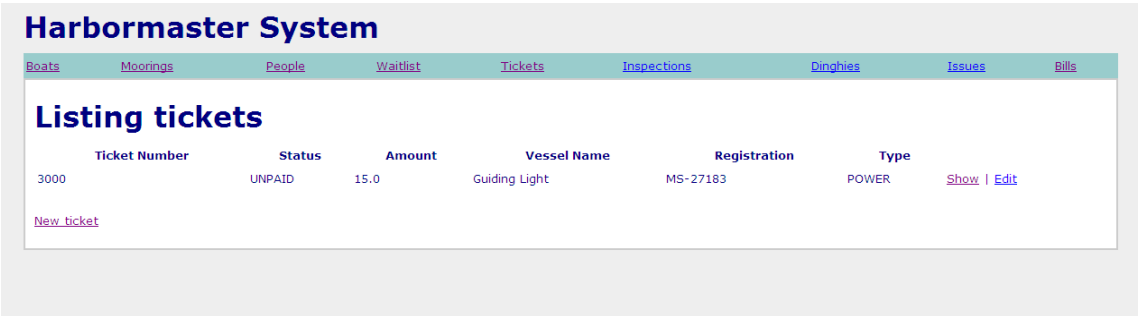

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Listing tickets
Ticket Number Status Amount Vessel Name Registration Type
3000 UNPAID 15.0 Guiding Light MS-27183 POWER Show | Edit
New ticket

Figure 29: Listing Tickets

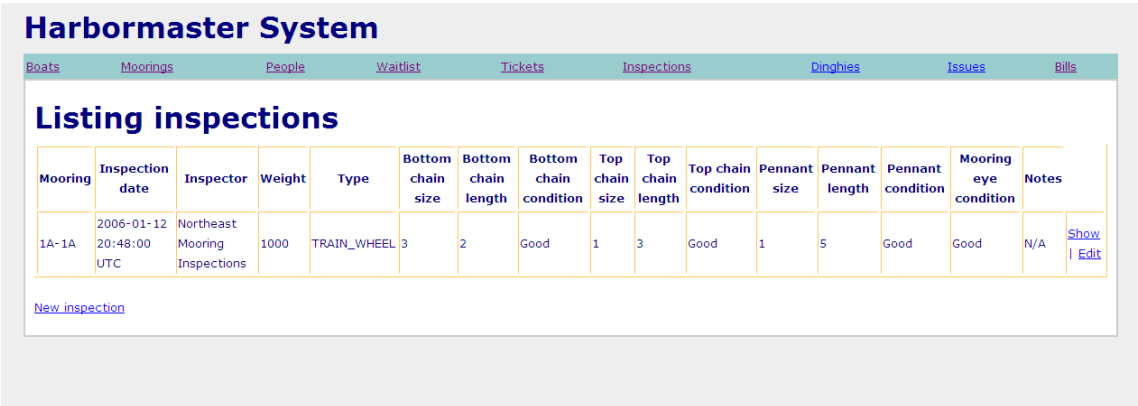

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Listing inspections
Mooring Inspection date Inspector Weight Type Bottom chain size Bottom chain length Bottom chain condition Top chain size Top chain length Top chain condition Pennant size Pennant length Pennant condition Mooring eye condition Notes
1A-1A 2006-01-12 20:48:00 UTC Northeast Mooring Inspections 1000 TRAIN_WHEEL 3 2 Good 1 3 Good 1 5 Good Good N/A Show | Edit
New inspection

Figure 30: Listing Inspections

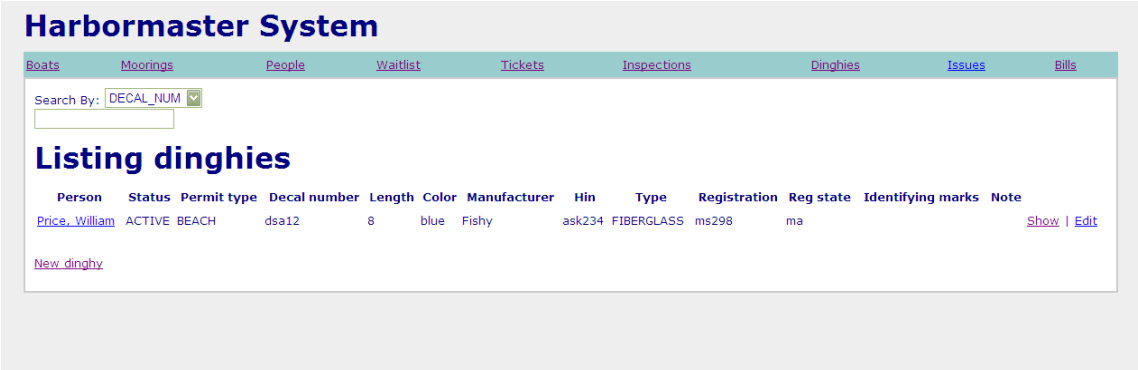

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Search By: DECAL_NUM
Listing dinghies
Person Status Permit type Decal number Length Color Manufacturer Hin Type Registration Reg state Identifying marks Note
Price, William ACTIVE BEACH dsa12 8 blue Fishy ask234 FIBERGLASS ms298 ma Show | Edit
New dinghy

Figure 31: Listing Dinghies

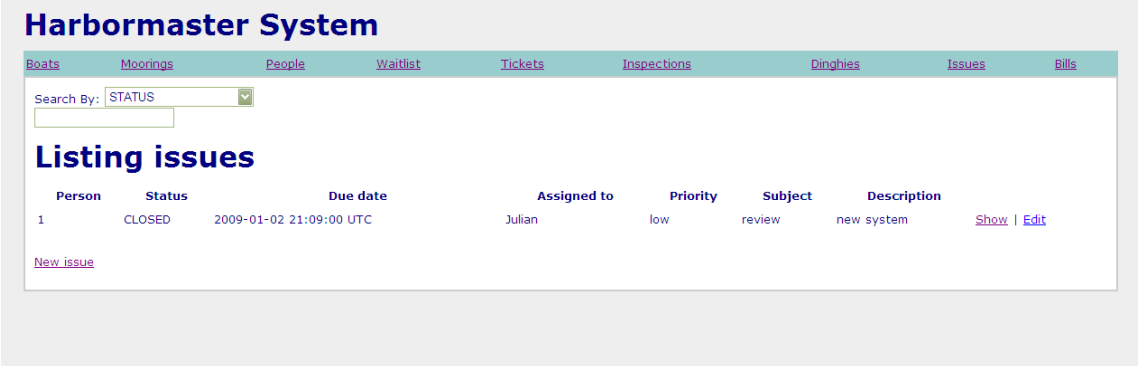

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Search By: STATUS
Listing issues
Person Status Due date Assigned to Priority Subject Description
1 CLOSED 2009-01-02 21:09:00 UTC Julian low review new system Show | Edit
New issue

Figure 32: Listing Issues

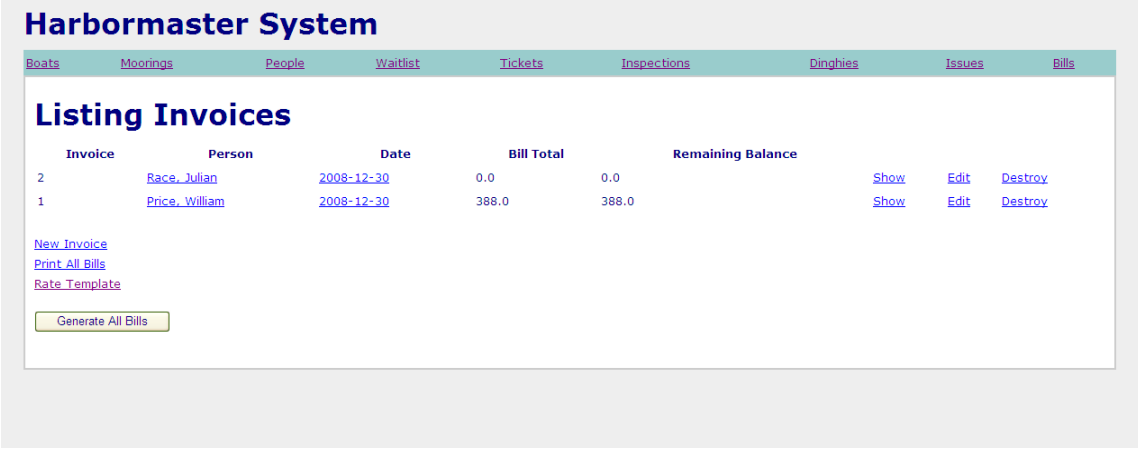

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Listing Invoices
Invoice Person Date Bill Total Remaining Balance
2 Race, Julian 2008-12-30 0.0 0.0 Show Edit Destroy
1 Price, William 2008-12-30 388.0 388.0 Show Edit Destroy
New Invoice
Print All Bills
Rate Template
Generate All Bills

Figure 33: Listing Bills

6.1.3 Adding a person

By filling in the fields below and clicking the “Create” button a new person is added into the database. This action is related to 3.1.2.1 Use Case #1: Enter New Person.

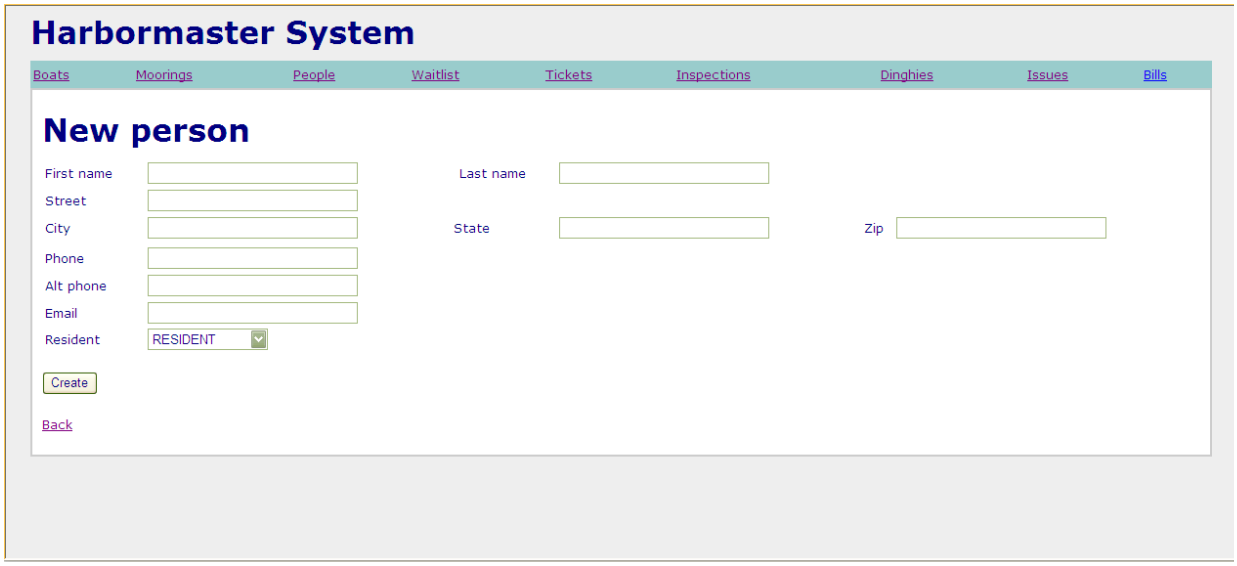


Figure 34: Adding a new person

6.1.4 Adding a Candidate to the Waitlist for Moorings

To add a person's request for a mooring to the waitlist, use the new waitlist screen. On this screen, a user must select a person from the drop-down box, in turn activating the boat drop-down to select a boat for the person. Listed next to the boat name is an id number to be entered into the next textbox. Select a status for the waitlist application and type in a desired location. This action is related to 3.1.2.2 Use Case #2: Add a Candidate to waitlist.

Figure 35: Add a candidate to waitlist

6.1.5 Adding a Boat

By filling in the fields below, selecting a boat type, pump-out side, owner, and status and clicking the "Create" button a new boat is added into the database. This action is related to 3.1.2.3 Use Case #3: Add a New Boat.

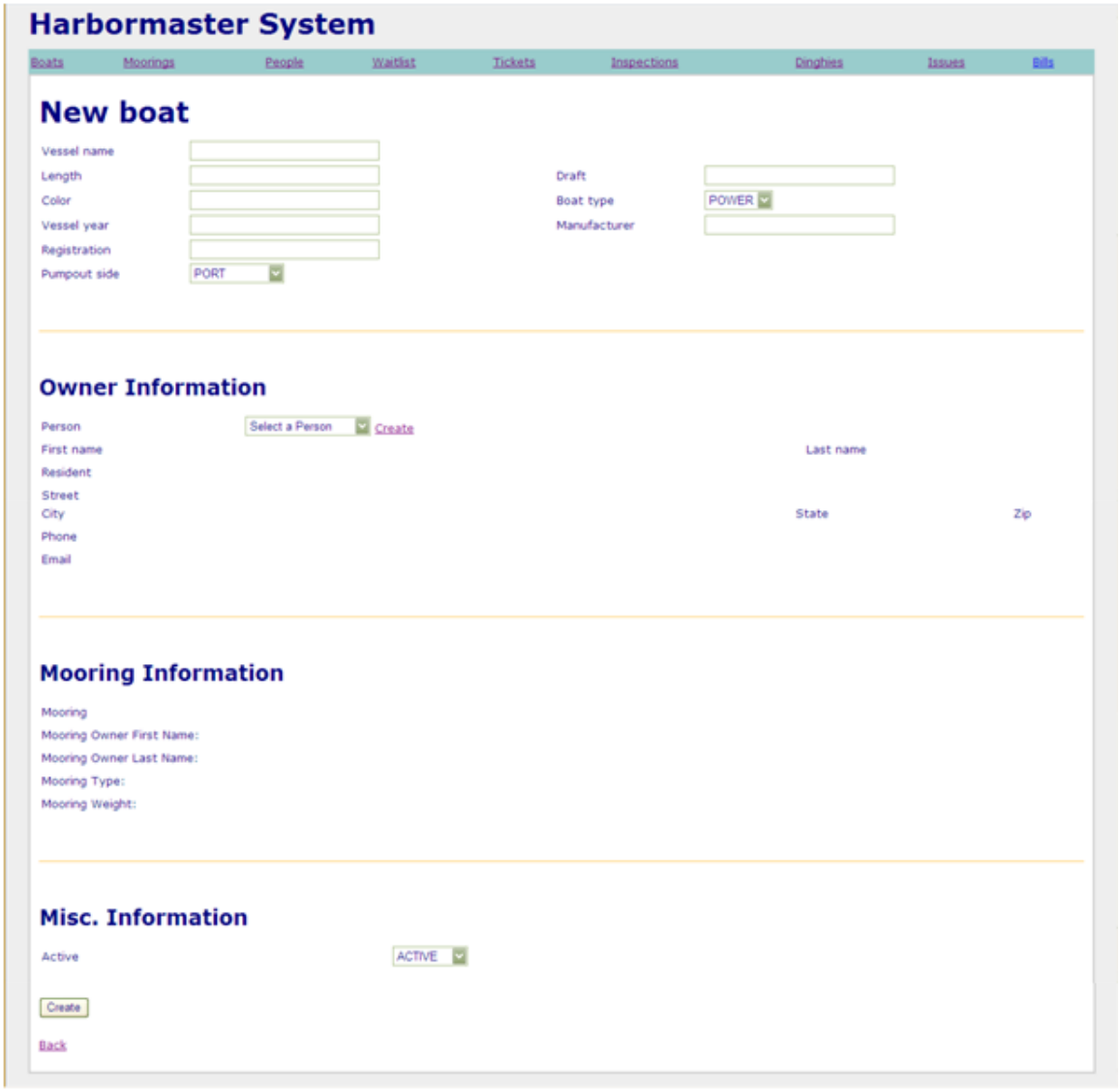


Figure 36: Adding a new boat

6.1.6 Adding a Mooring

By filling in the fields below, selecting a person, a boat and clicking the “Create” button a new mooring is added into the database. This action is related to3.1.2.4 Use Case #4: Register a New Mooring.

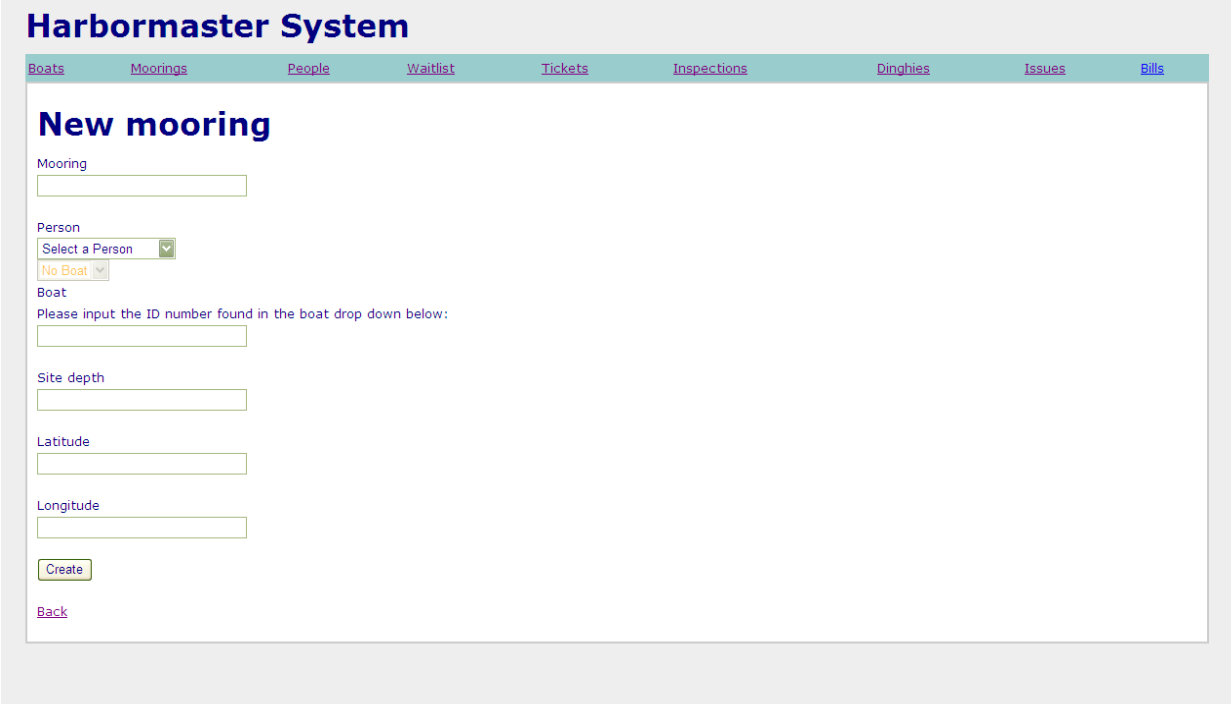


Figure 37: Adding a new mooring

6.1.7 Adding a New Dinghy

By filling in the fields below, selecting a person, status, location, permit type, dinghy type and clicking the “Create” button a new dinghy is added to the system. This action is related to 3.1.2.5 Use Case #5: Add New Dinghy.

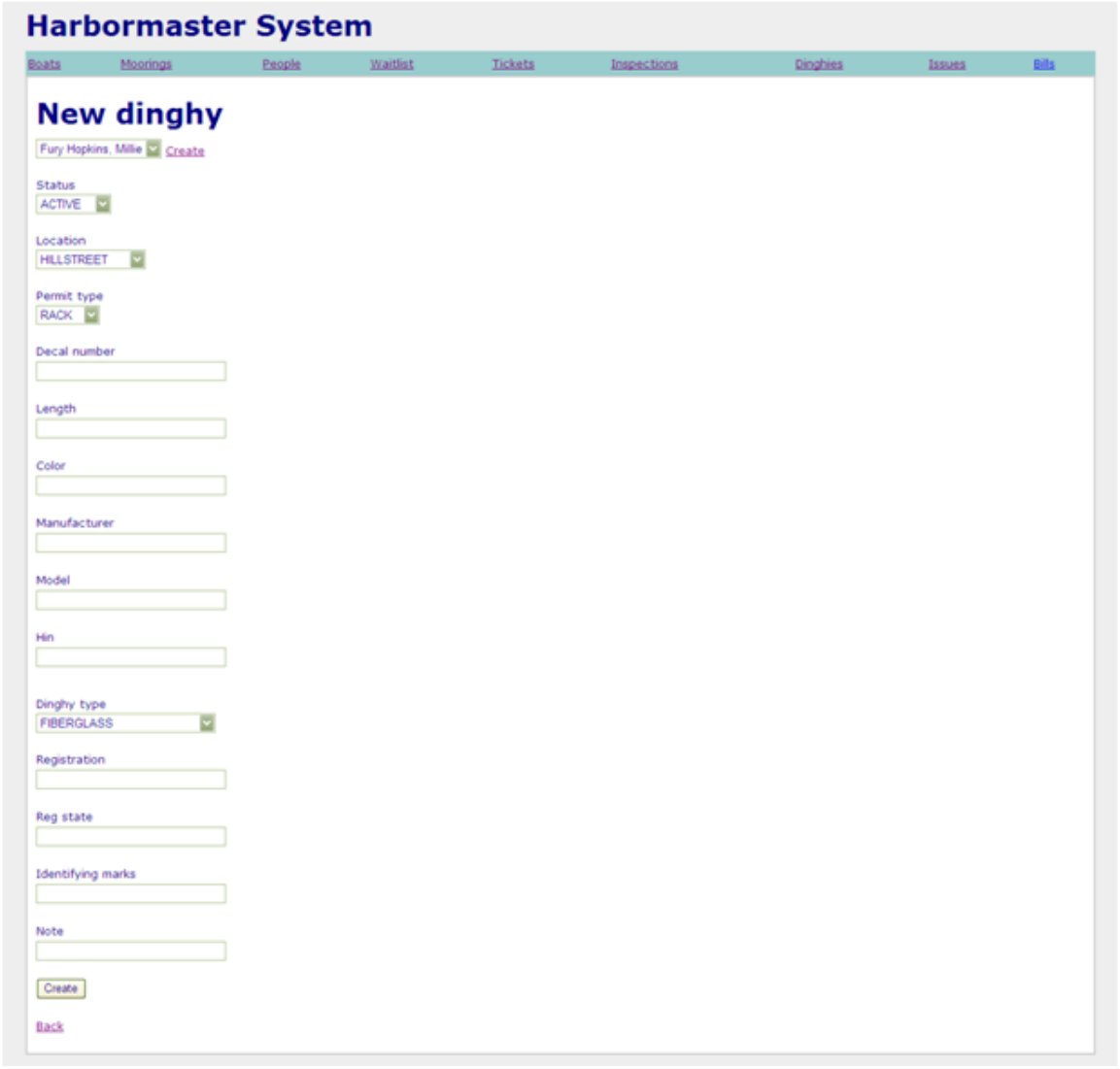


Figure 38: Adding a new dinghy

6.1.8 Adding a New Issue

By filling in the fields below, selecting a person, status, due date and clicking the "Create" button a new issue is added to the system. This action is related to 3.1.2.6 Use Case #6: Add New Issue.

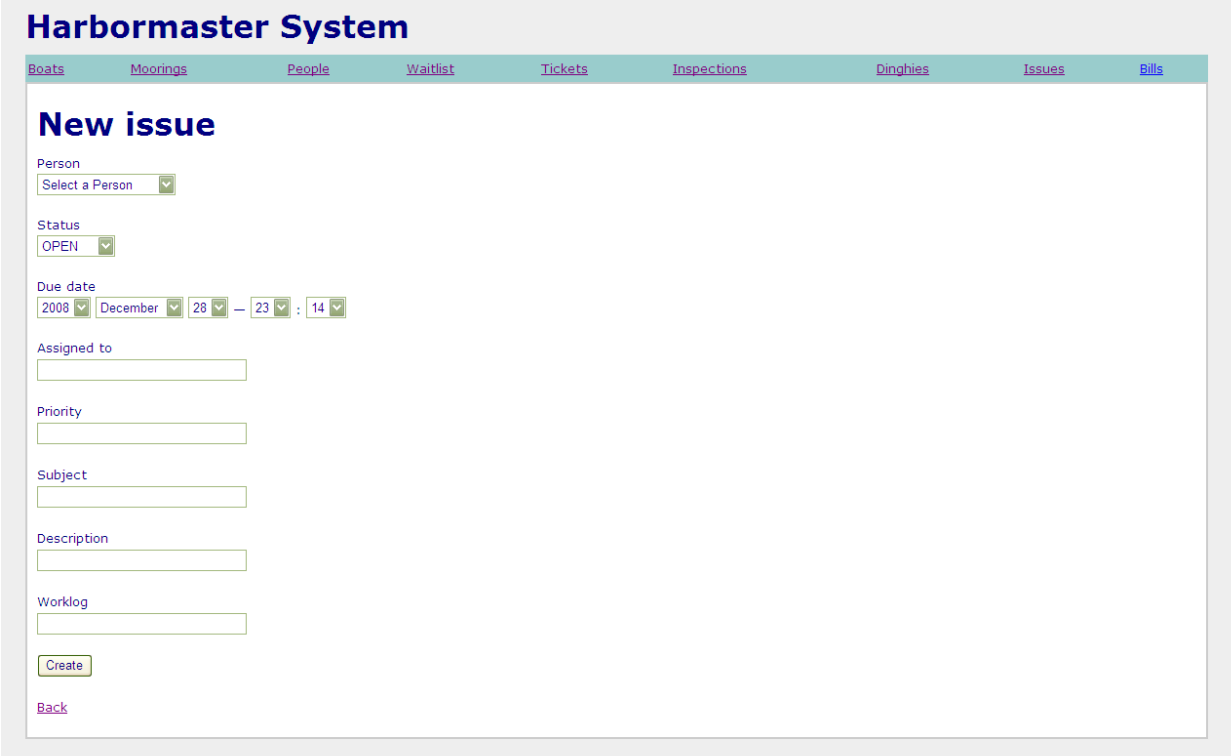


Figure 39: Adding a new issue

6.1.9 Adding a New Inspection

By filling in the fields below, selecting an inspection date, type and clicking the “Create” button a new inspection is added to the system. This action is related to 3.1.2.11 Use Case #11: Add an Inspection.

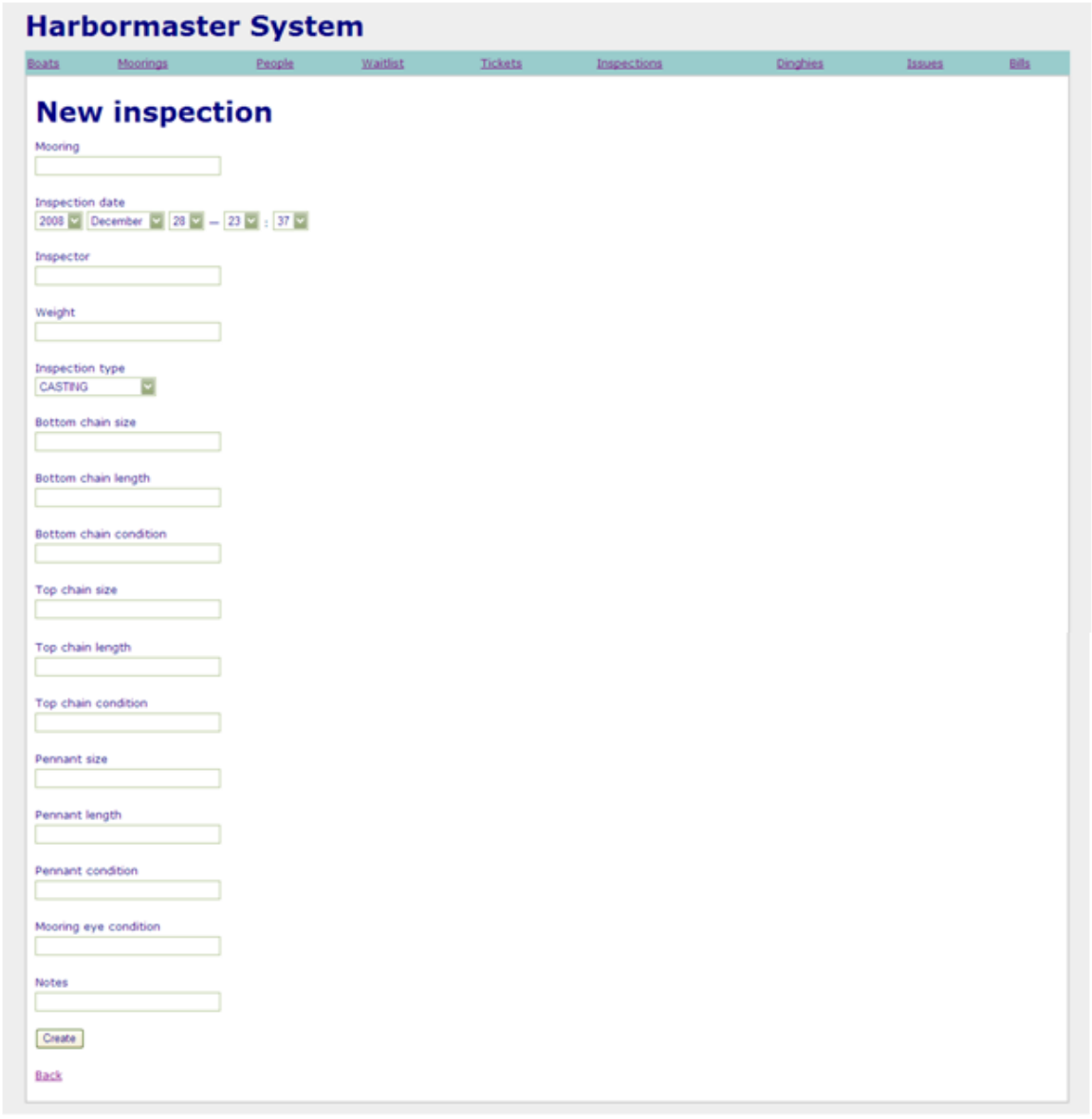


Figure 40: Adding a new inspection

6.1.10 Editing a mooring

By changing any of the fields below and clicking the “Update” button an existing mooring is changed in the system. This action is related to 3.1.2.13 Use Case #13: RFP: Change in Mooring Permit.

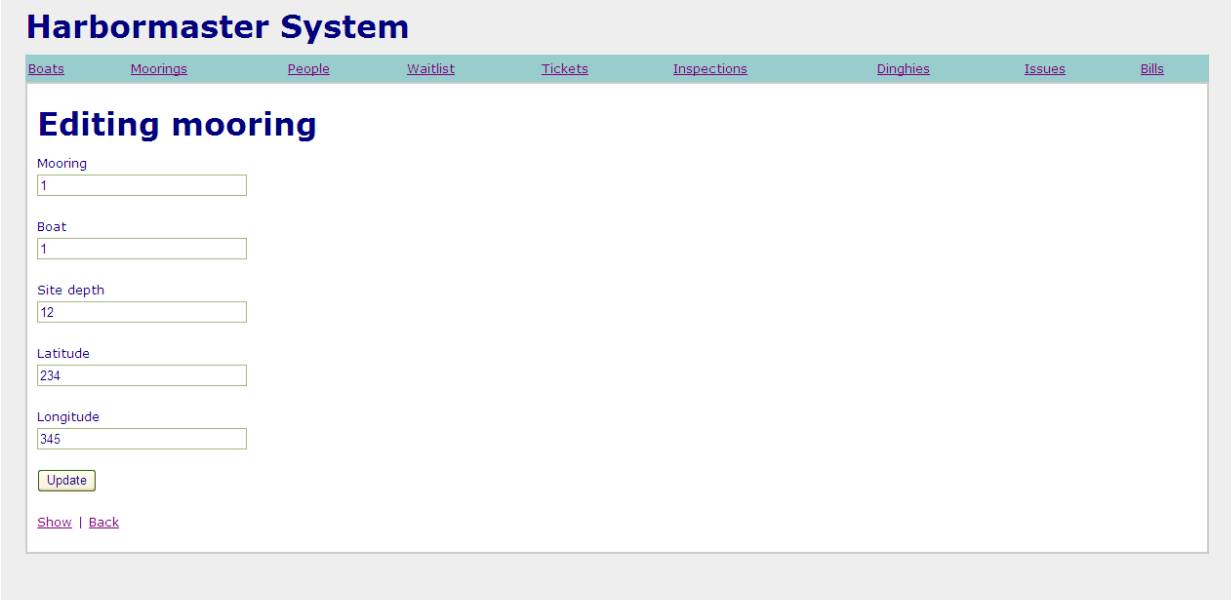


Figure 41: Editing a mooring

6.1.11 Adding a New Ticket

Filling out the following form, selecting a person, boat, status, a date issued and clicking the “Create” button adds a new ticket to the system. This action is related to 3.1.2.15 Use Case #15: Add a New Ticket.

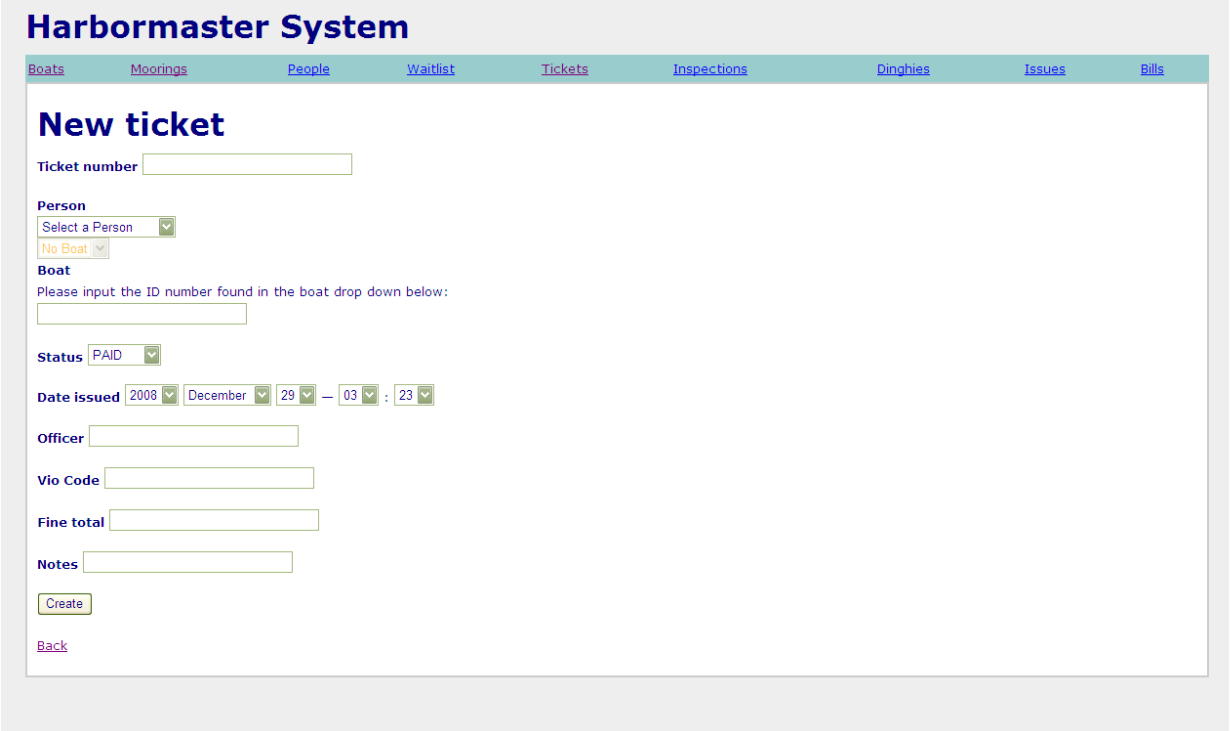


Figure 42: Adding a new ticket

6.1.12 Ticket Payment

By changing the “Status” drop down to “Paid” in the Editing ticket screen and clicking the “Update” button a ticket is now recorded as paid in the system. This action is related to 3.1.2.16 Use Case #16: Ticket Payment.

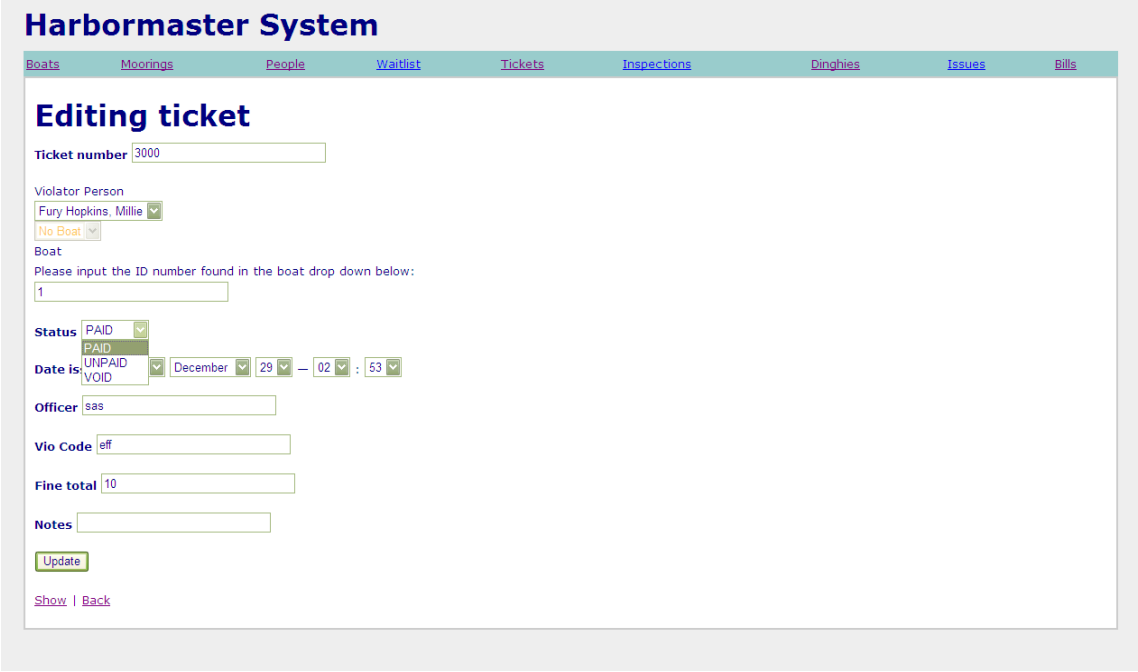


Figure 43: Ticket payment

6.1.13 Adding Rate Information

To provide historical data on payment rates, a new rate must be entered for each category each year. By filling in the fields below, selecting a rate type and clicking the "Create" button, a rate is added to the system.

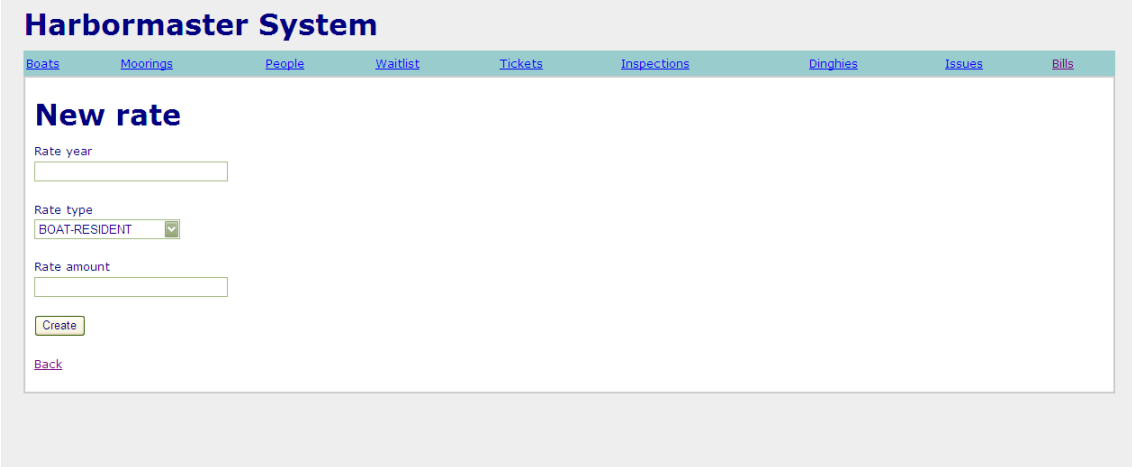


Figure 44: Input Rate Information

6.1.13 Listing Rate Information

The page below shows the rates listed in the system. Bills for the current year are based on the current year's rate information.

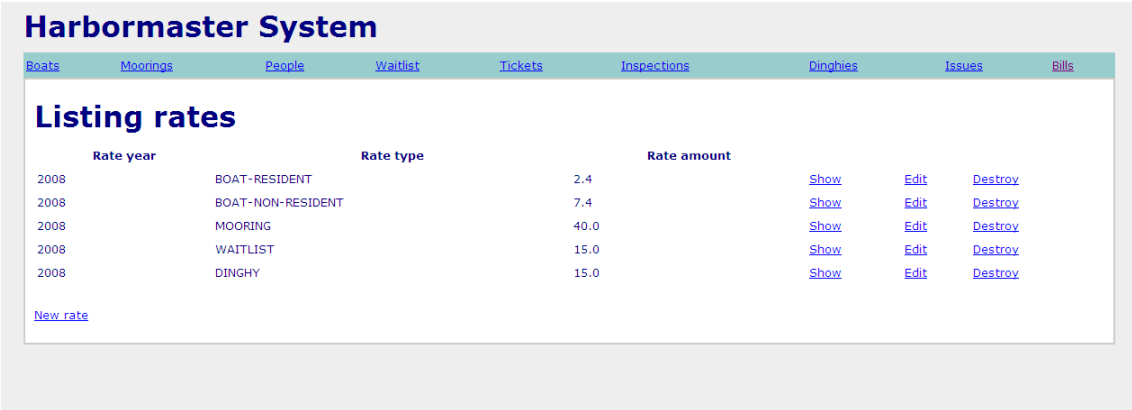


Figure 45: Listing Rate Information

6.1.14 Editing Rate Information

By changing the fields below and clicking the "Update" button, the user is able to edit rate information for a particular rate type or year.

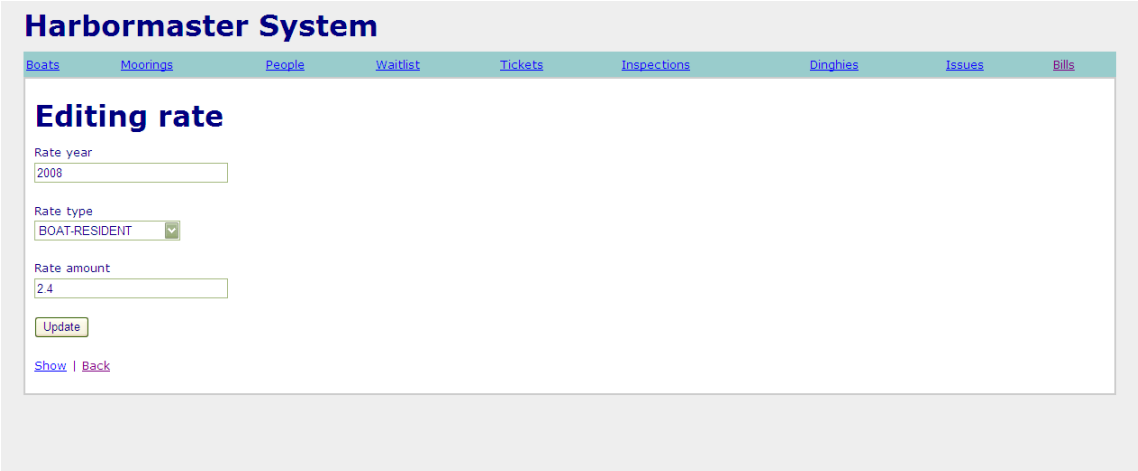


Figure 46: Editing Rate Information

6.1.15 Generating a Bill

Proofing the bills generated by the system ensures billing line items and customer information is correct. From invoice pages the user is able to print a bill for this specific customer. This action is related to 3.1.2.10 Use Case #10: Generate Yearly Billing Process.

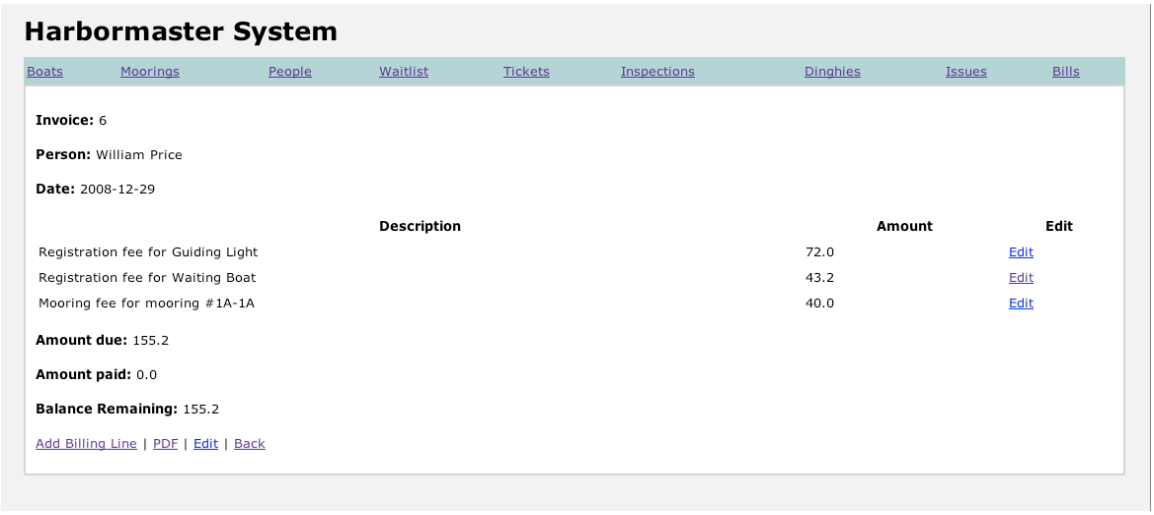

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Invoice: 6
Person: William Price
Date: 2008-12-29
Description
Amount
Edit
Registration fee for Guiding Light 72.0 Edit
Registration fee for Waiting Boat 43.2 Edit
Mooring fee for mooring #1A-1A 40.0 Edit
Amount due: 155.2
Amount paid: 0.0
Balance Remaining: 155.2
Add Billing Line | PDF | Edit | Back

Figure 47: Generate a bill

6.1.15 Editing Pages

In addition to the listing and new pages for each data type in the system, there are also related edit pages for changing data. The following are screenshots of the edit pages for pages for Boats, People, Waitlist, Inspections, Dinghies, Issues and Bills.

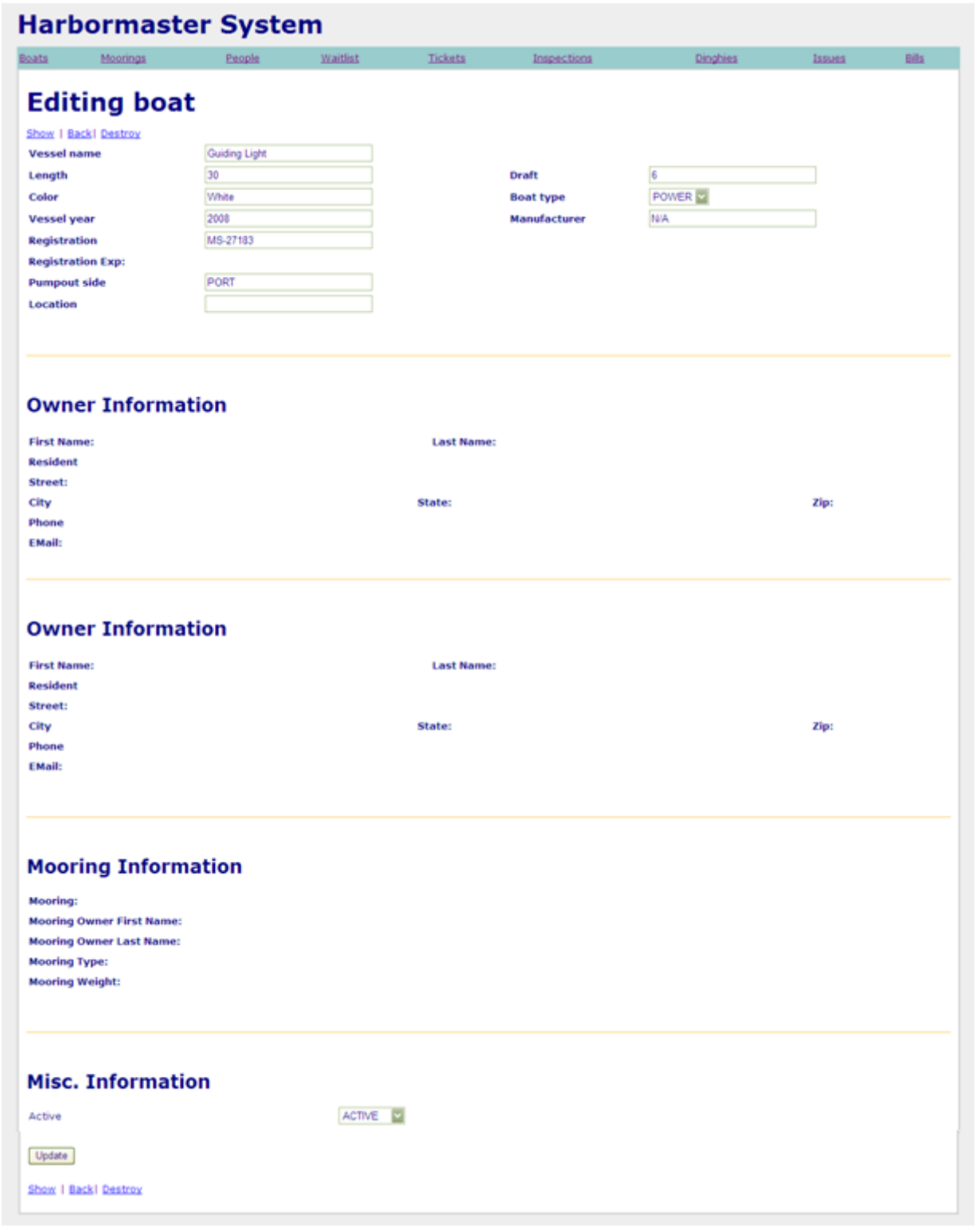


Figure 48: Edit Boat Page

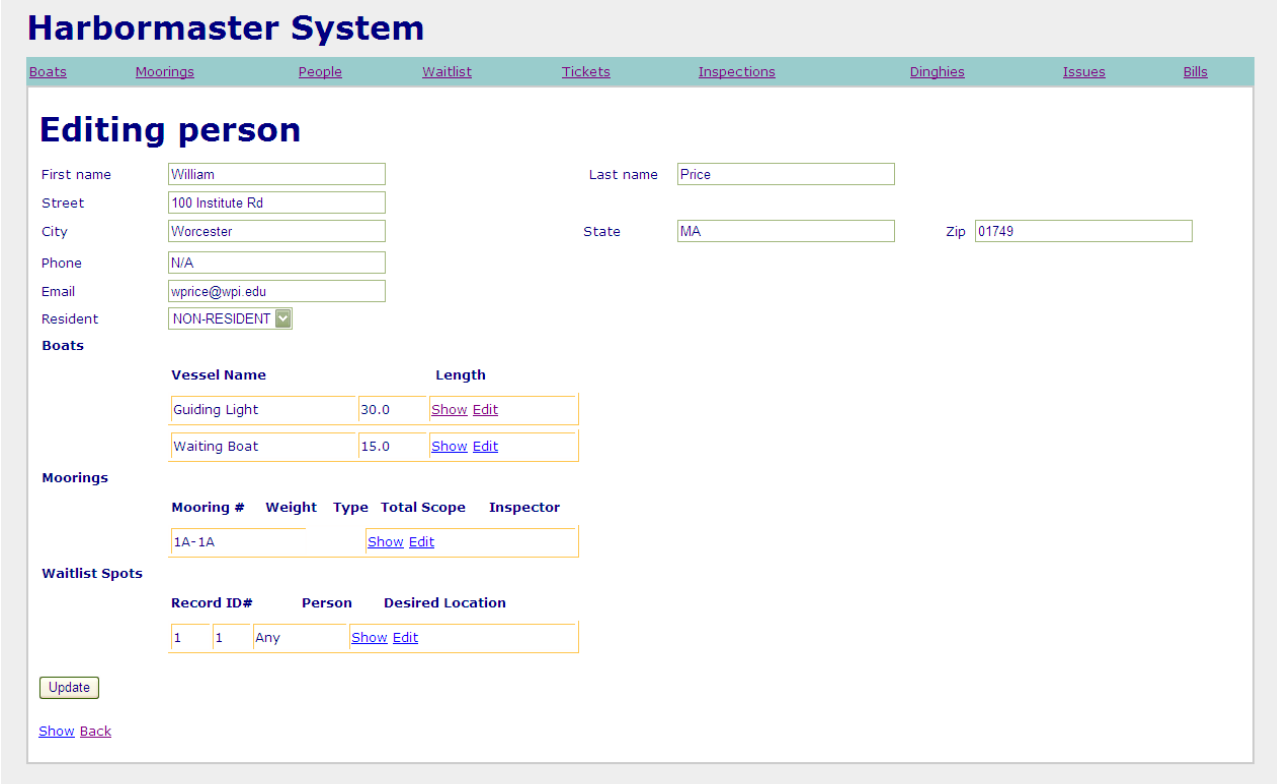


Figure 49: Edit Person Page

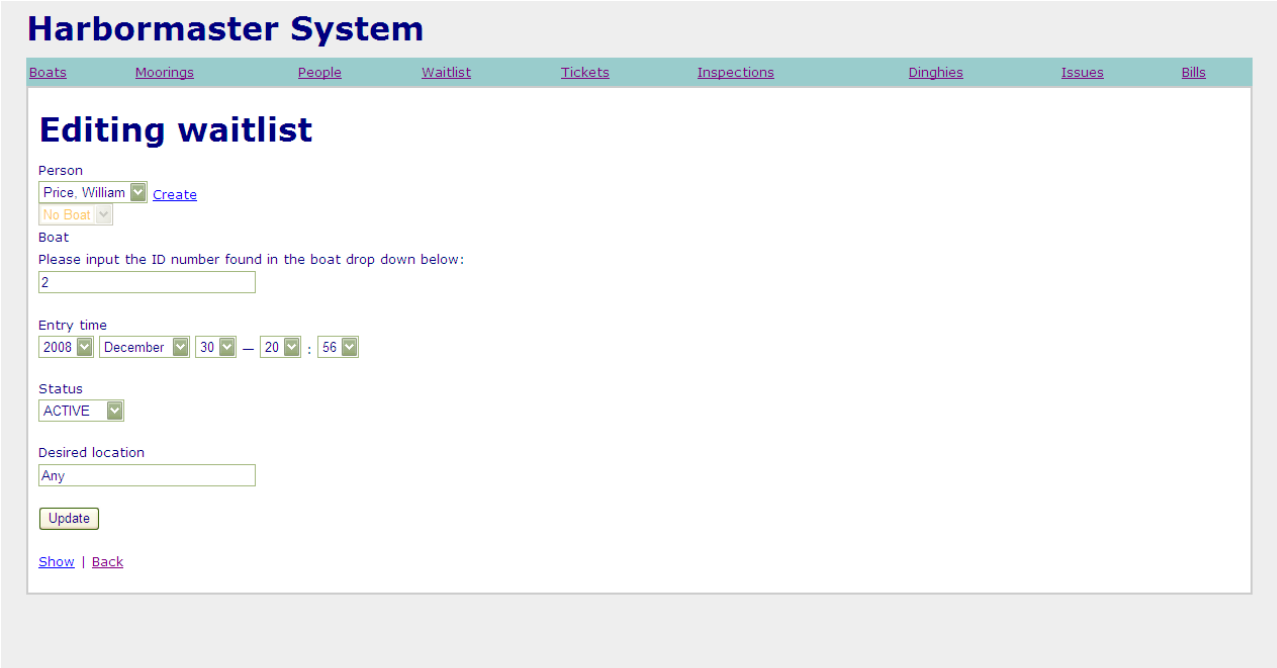


Figure 50: Edit Waitlist Page

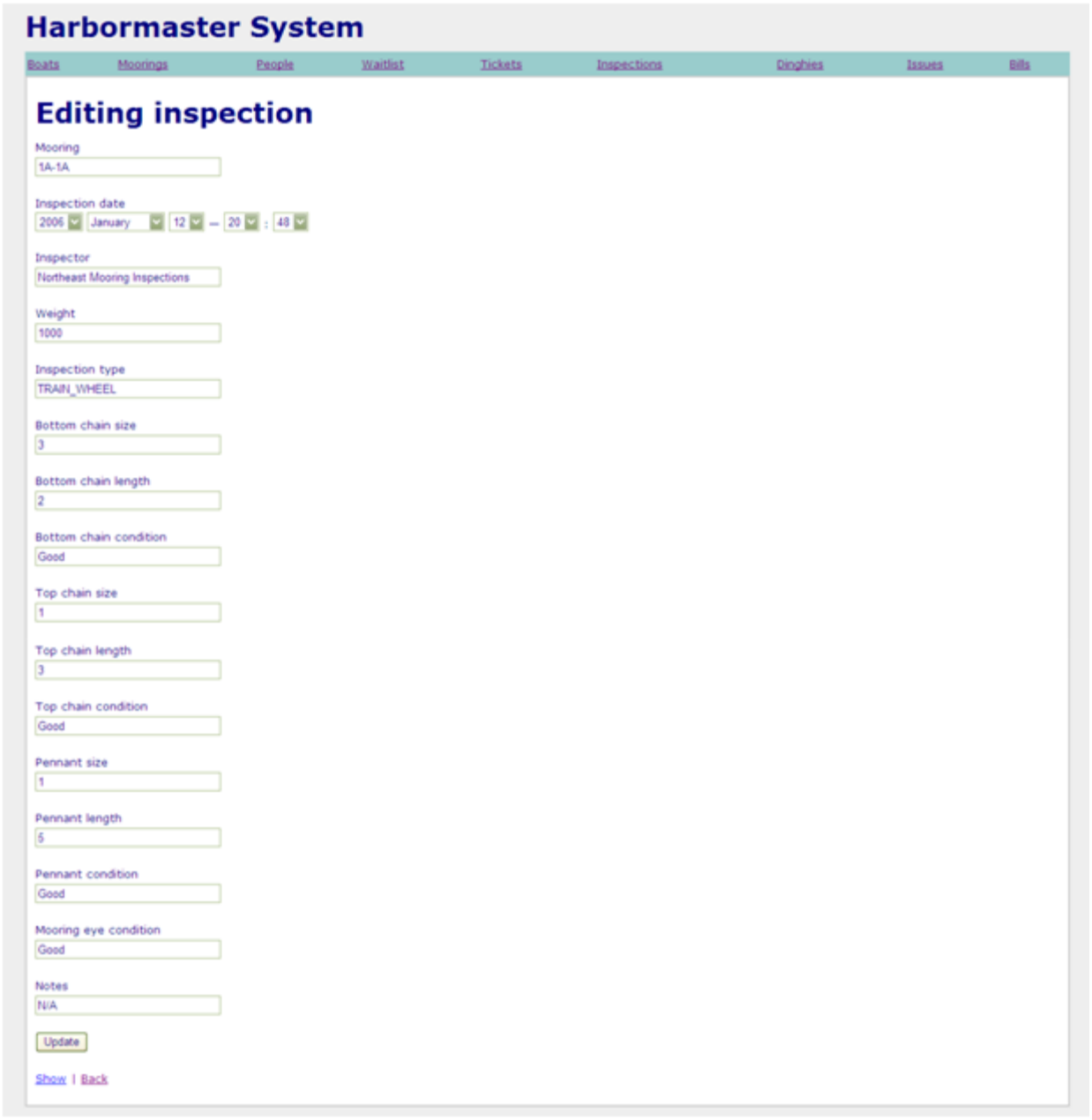


Figure 51: Edit Inspection Page

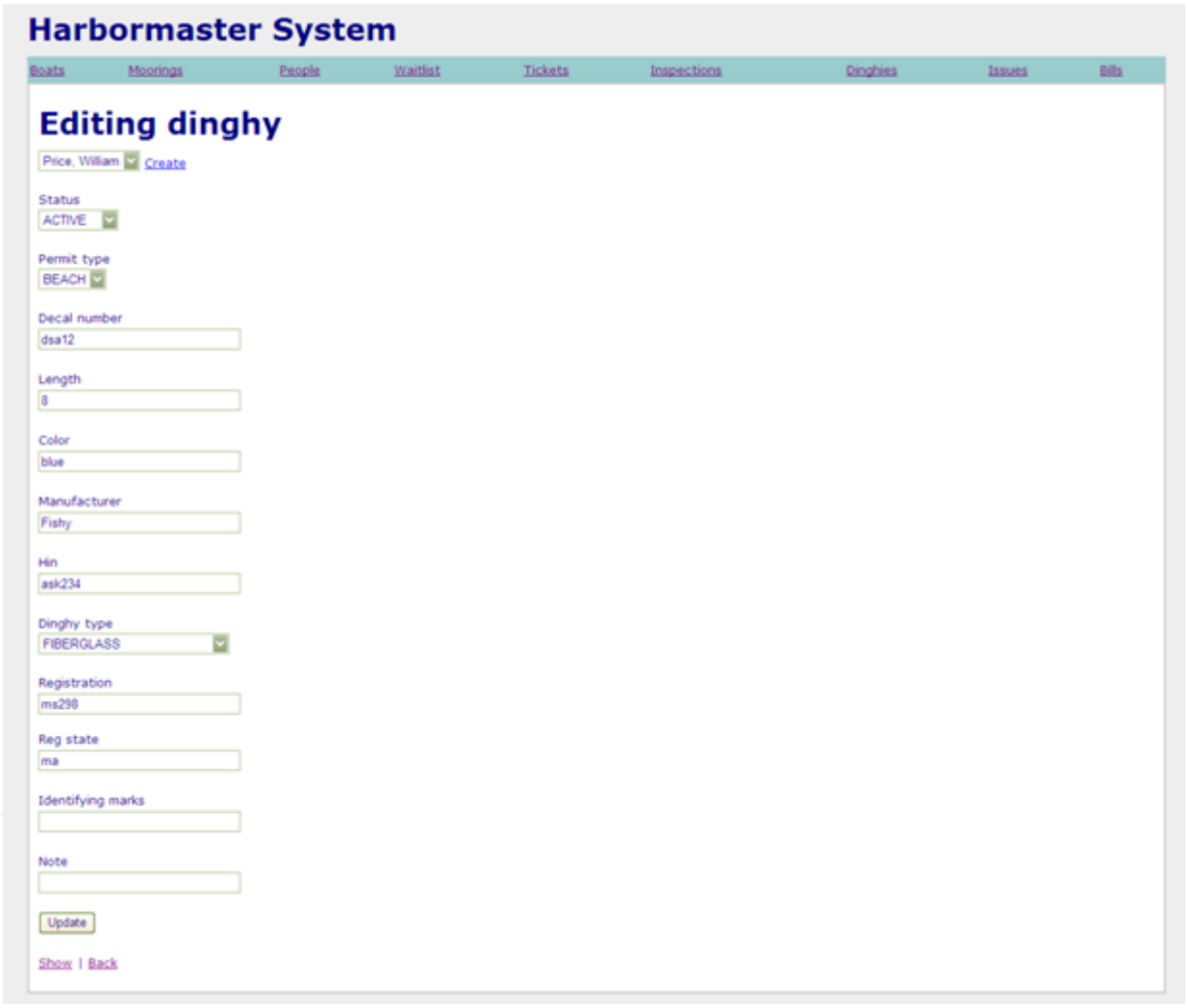


Figure 52: Edit Dinghy Page

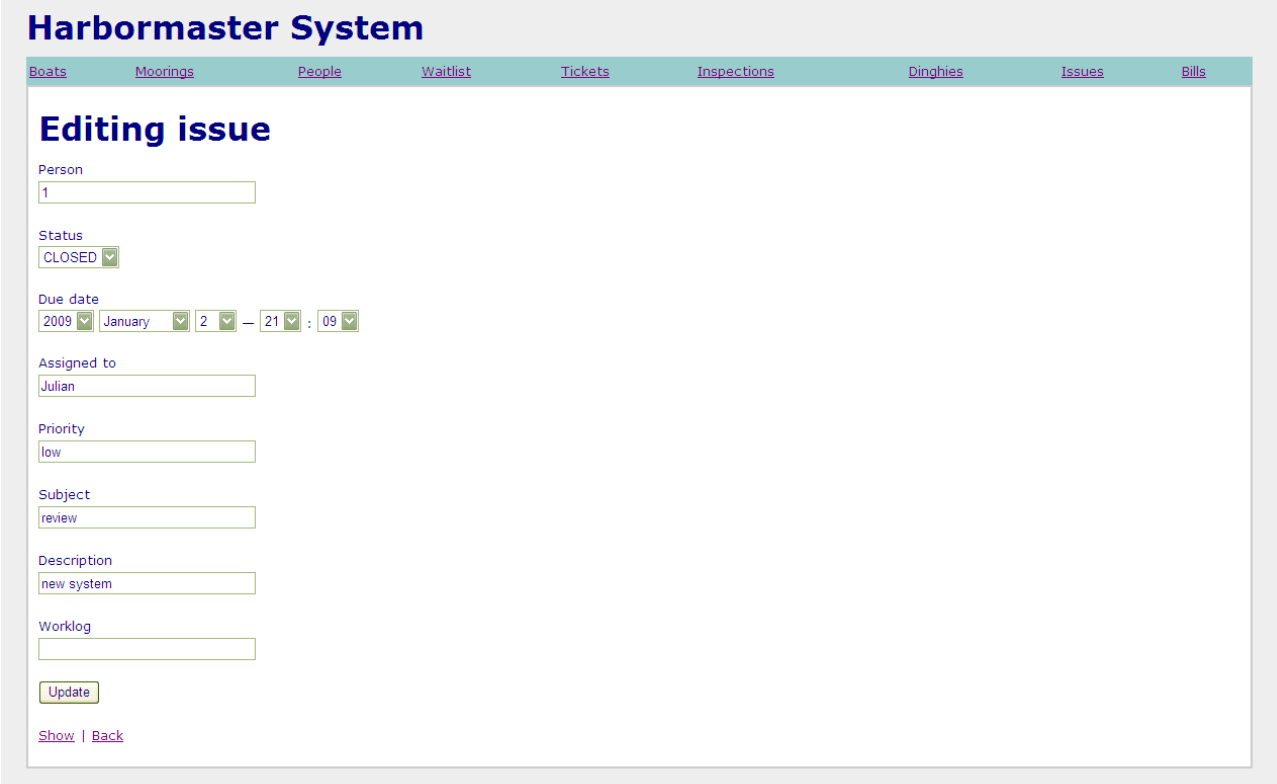


Figure 53: Edit Issue Page

Figure 54: Edit Invoice Page

6.1.16 Show Pages

To provide the ability to view data within the system there are "show" pages. By clicking on the "show" link beside any record on the listing pages the user is able to view that specific record in the system. Many of the show pages pull information from other entities to show the relationship between them. For example, by looking at the show page for a boat, if that boat has a mooring, the mooring information also shows as does the owner information.

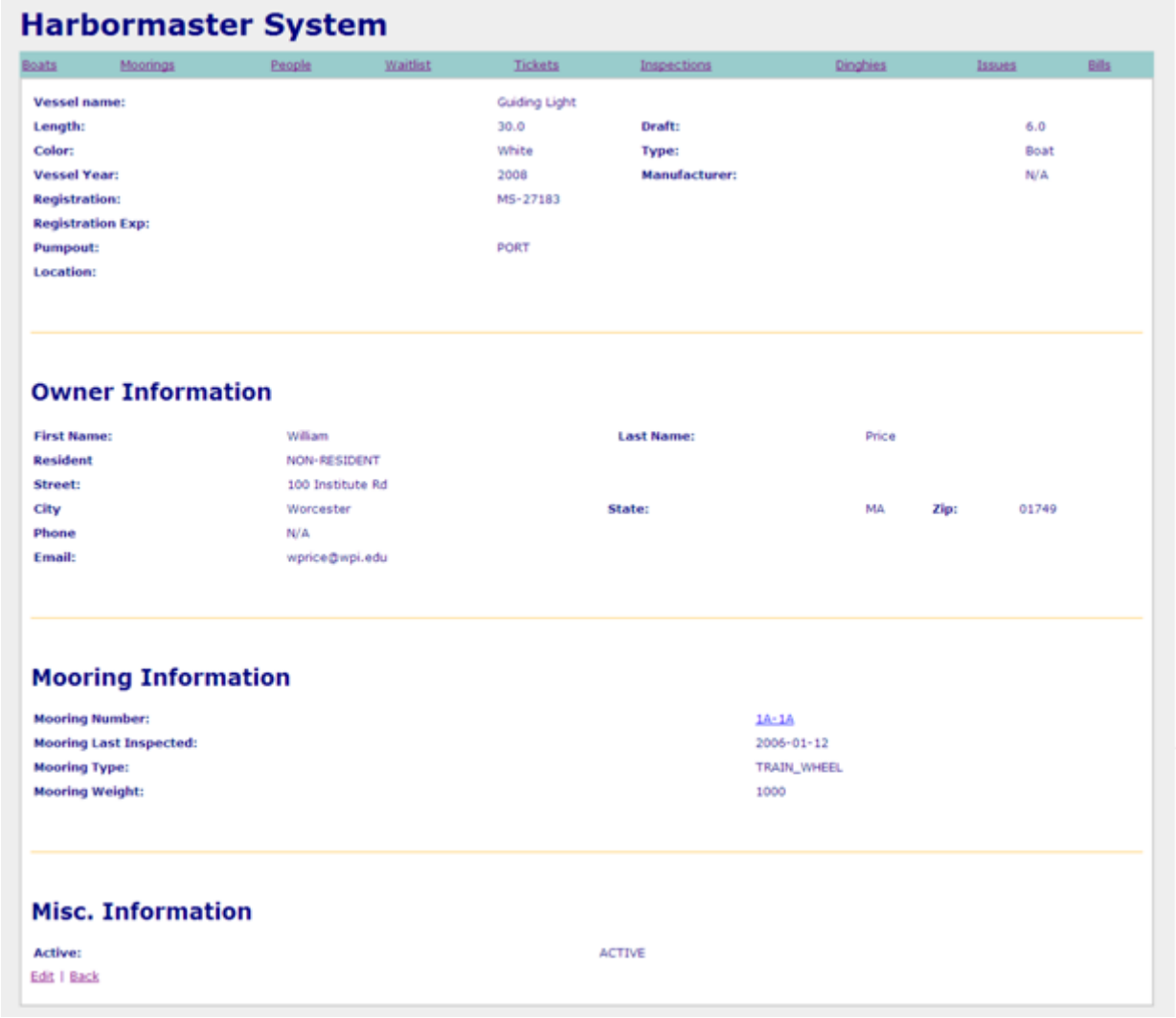


Figure 55: Show Boat Page

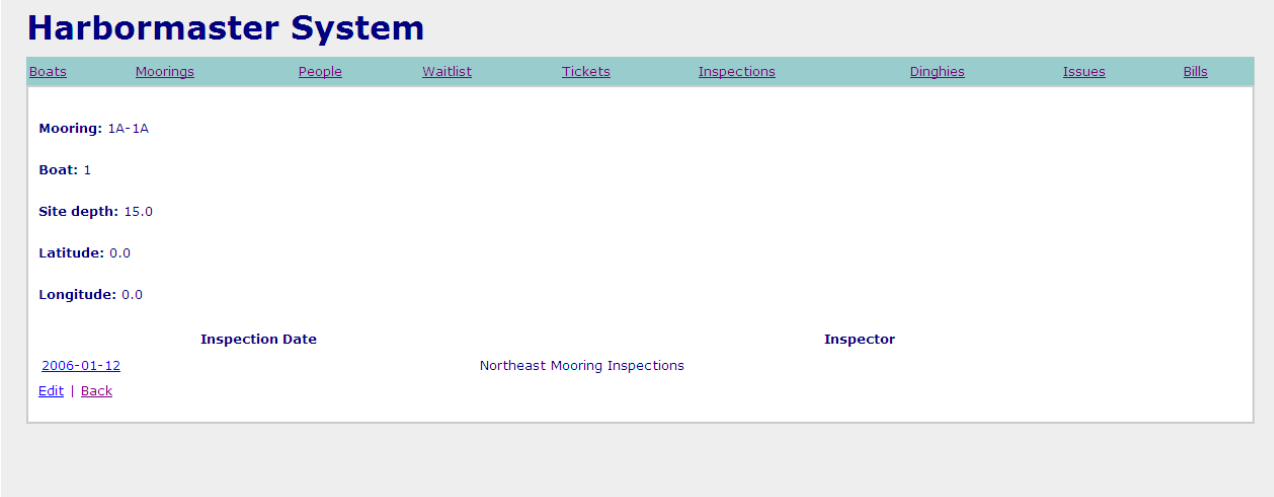


Figure 56: Show Mooring Page

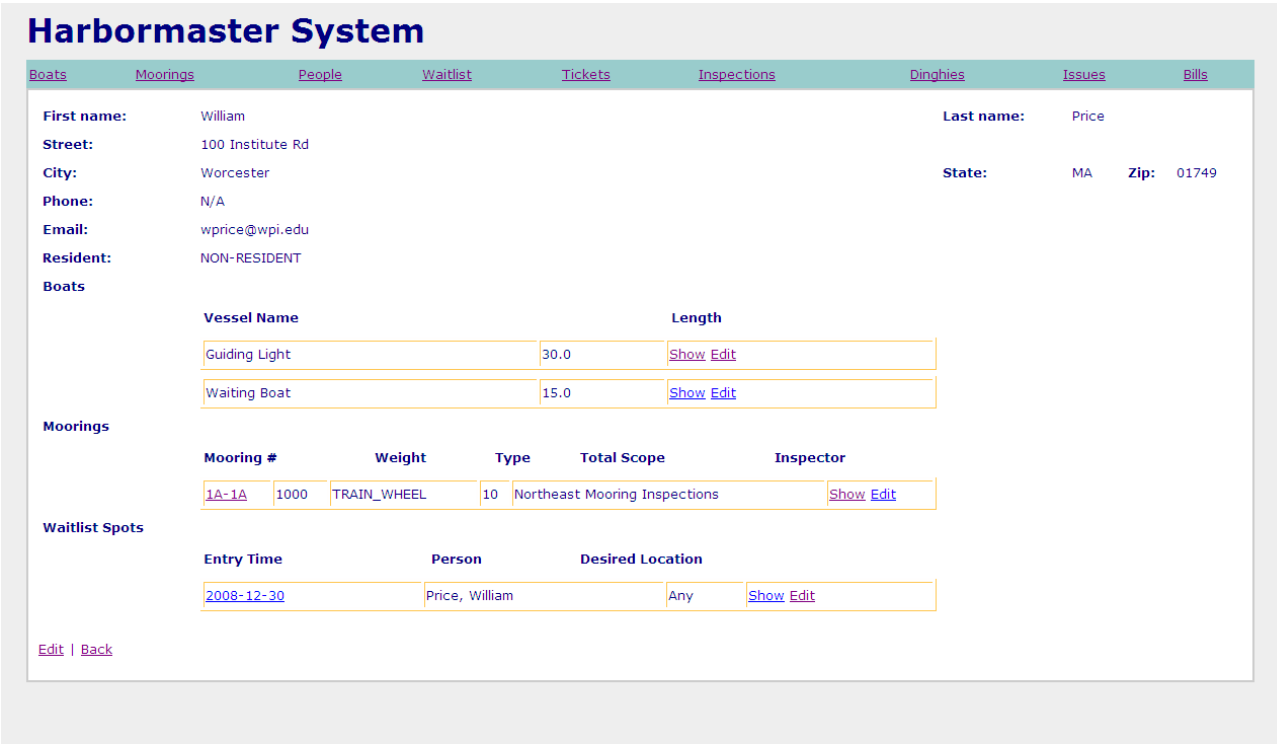


Figure 57: Show Person Page

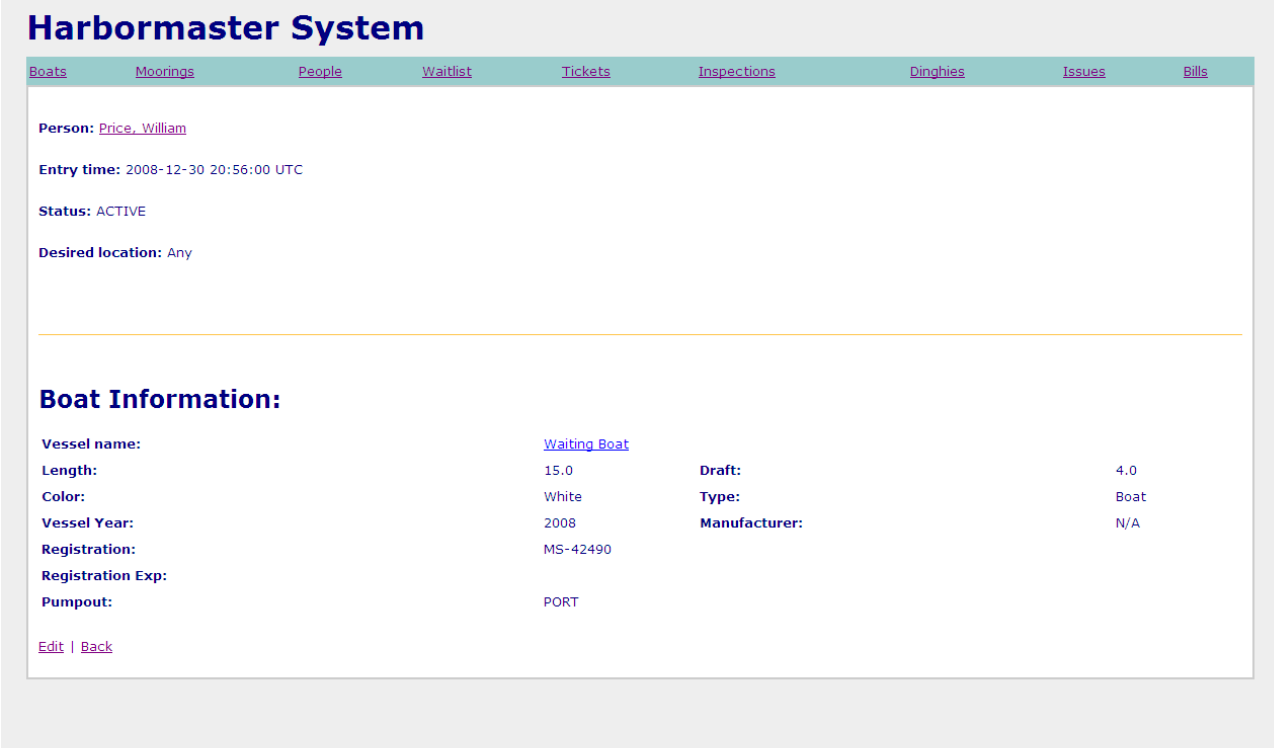


Figure 58: Show Waitlist Page

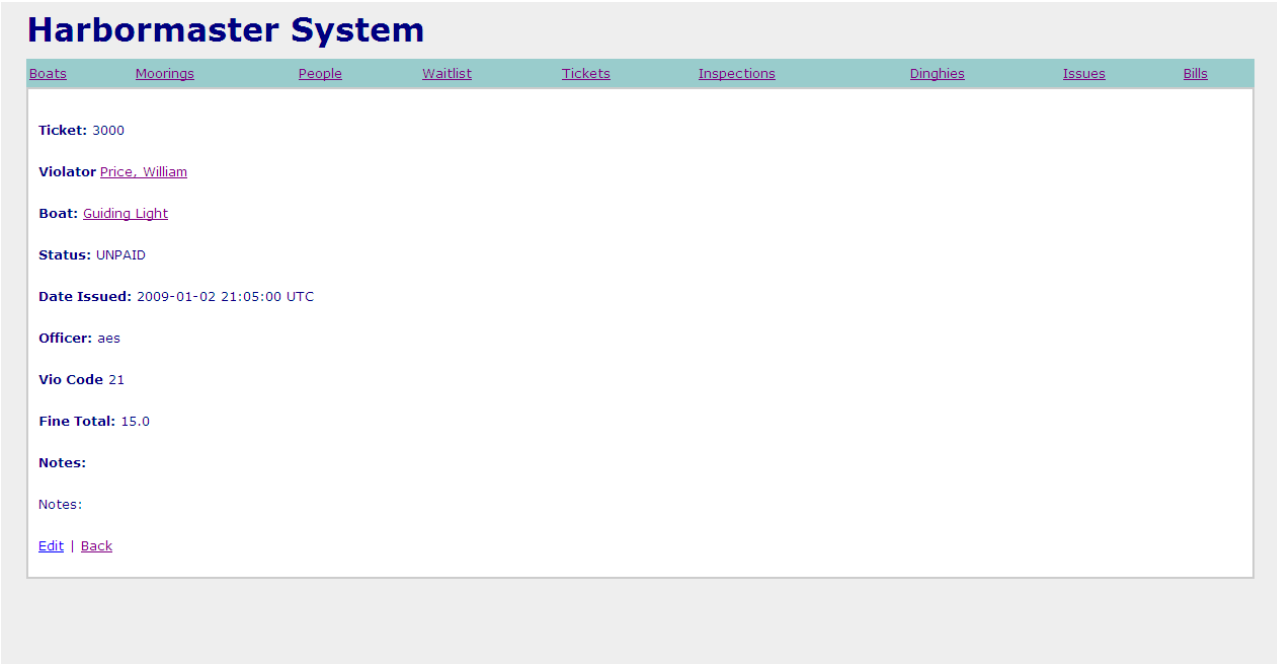


Figure 59: Show Ticket Page

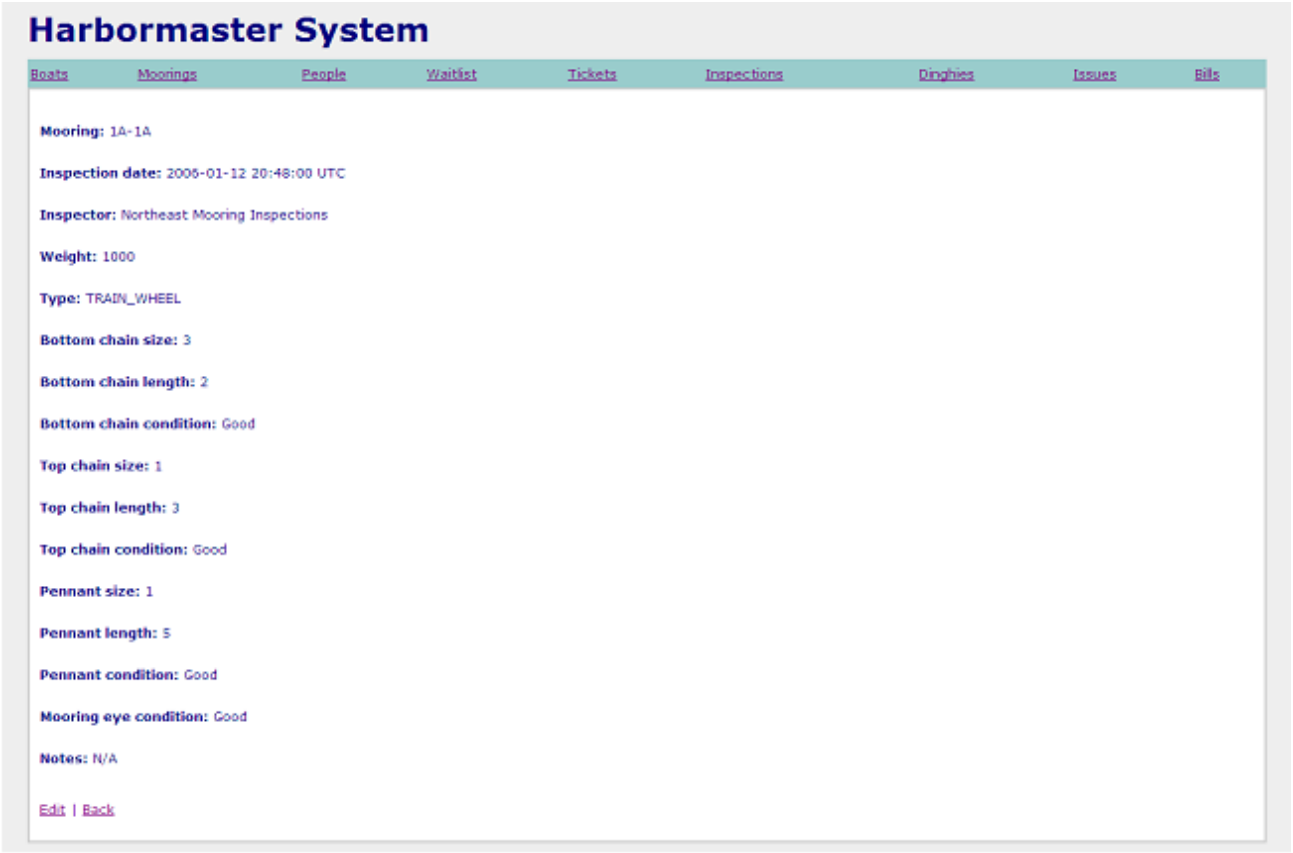


Figure 60: Show Inspection Page

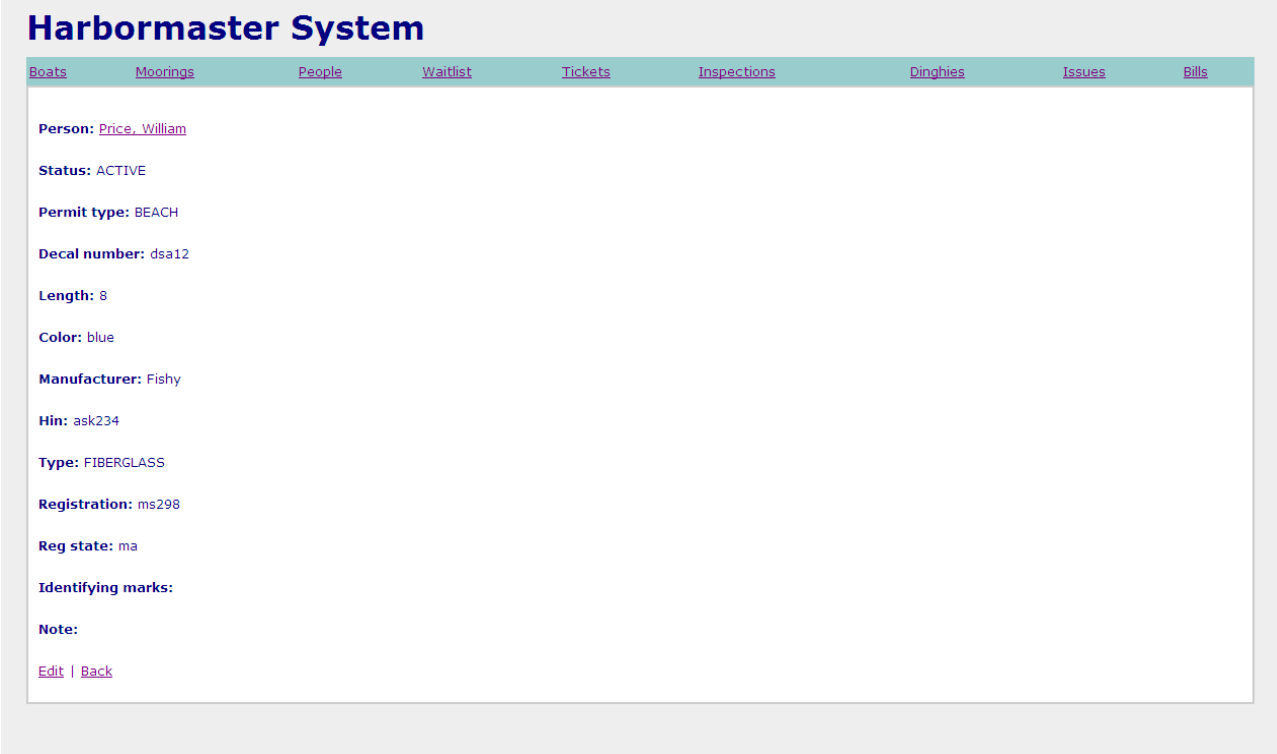


Figure 61: Show Dinghy Page

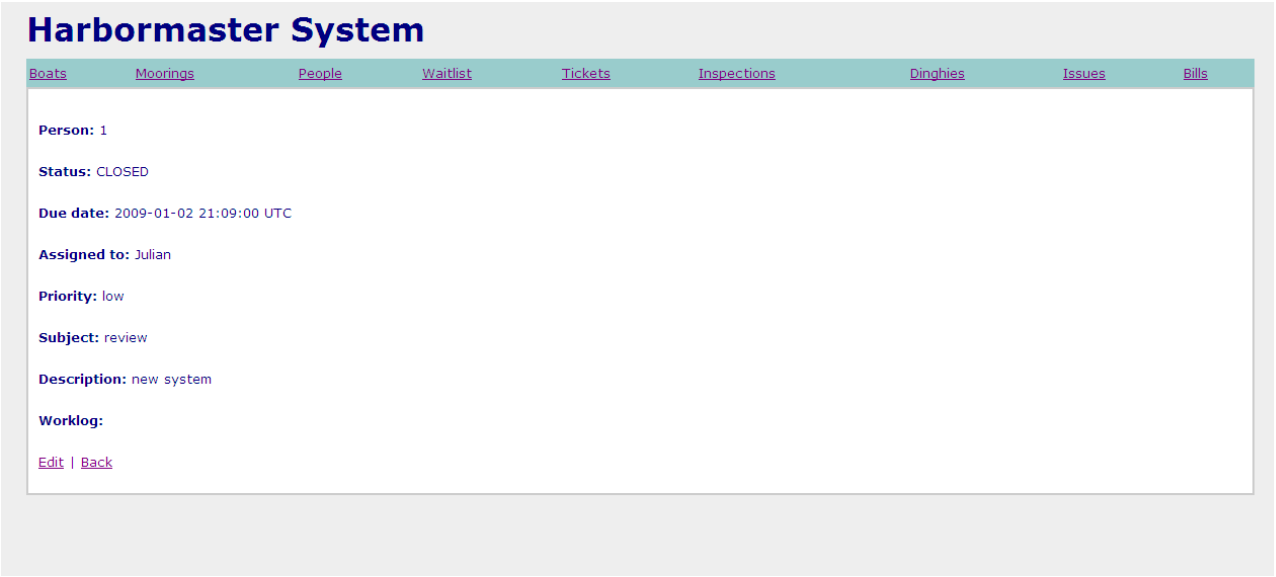


Figure 62: Show Issue Page

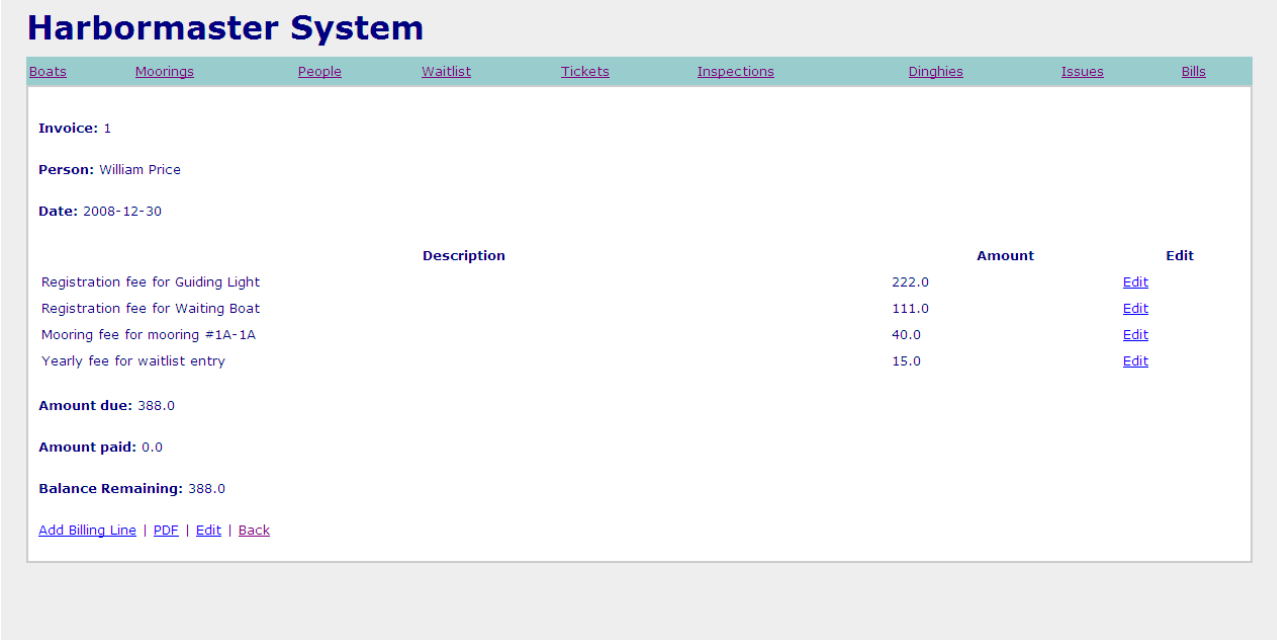

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Invoice: 1
Person: William Price
Date: 2008-12-30
Description
Amount
Edit
Registration fee for Guiding Light 222.0 Edit
Registration fee for Waiting Boat 111.0 Edit
Mooring fee for mooring #1A-1A 40.0 Edit
Yearly fee for waitlist entry 15.0 Edit
Amount due: 388.0
Amount paid: 0.0
Balance Remaining: 388.0
Add Billing Line | PDF | Edit | Back

Figure 63: Show Invoice Page

6.2 Additional References

For additional information about using the system please see Appendix IV. User Manual. For more information on the construction of the website or how the web framework works, please reference Appendix II. Technical Documentation.

Chapter 7. Conclusions

7.1 Testing

Two approaches were used to the testing of this system. As the code was developed, each developer performed unit tests of their own code. No code was acceptable until it passed the unit tests satisfactorily on the developer's local web server. Each field on each webpage was tested as it was added to the application, and all functions were tested before they were committed to the repository.

The second approach used in the system testing involved having the developer who did not write the code test it on their machine. This allowed a fresh perspective, and also helped us ensure that the code worked across platforms, as one developer used a UNIX-based machine to develop, and another used Windows. Having each developer perform unit tests for the pages helped ensure that nothing would slip through the cracks.

7.2 Deployment

The application has been deployed on the server located in the Dartmouth Harbormaster office. It is running on version 2.2 of Apache web server, and is accessible from any computer with Internet access by entering the following URL into a web browser:

http://dartmouthharbormaster.webhop.net

For more information on the deployment of the system, such as configuration settings, please reference the technical documentation in Appendix II.

7.3 Future Recommendations

While this system is functional and meets the requirements set forth at the beginning of this project, there are certain features that could be improved upon in the future. Mr. Race has submitted requests for some formatting and functionality changes that should be completed at a future date. These future additions will be broken into three sections: formatting, functionality, and deployment.

Formatting:

- On the new inspection page place chain sizes/lengths into a table format to make data entry more intuitive.
    - Rows: Chain location (Top/Bottom/Pennant)
    - Columns: Size, Length, Condition
- Put the values into a table on the rates index page
    - Rows: Year
    - Columns: Rate Types
- Alter formatting on printable PDF bills.
    - Place mailing address in location to line up with a windowed envelope.

Functionality:

- On all tables create a new column that will store the username of the last person to update the record.
- Create a function that will actively sort data on index pages based upon user input.
    - Example: A user clicks a link to sort all people by city without reloading.
- Investigate the live search functionality to remove case sensitivity from the searches.

Deployment:

- Use apache security to only allow permitted users to view the website.
- Consider hosting application on UNIX-based server.
    - Rails runs faster on a UNIX system.
- Consider upgrading server hardware to enhance performance.
    - Upgrades to:
        - Processor
        - RAM

7.3 Group Reflection

Over the course of the past four and a half months we have had the pleasure of working on a unique and challenging project. The experience has been very beneficial to both group

members, as we have gained invaluable experience in dealing directly with a client. In many career paths that we may choose to take at some point in our lives, we will need to work directly with a client.

Over our time at WPI we have completed a plethora of projects of varying scope and objectives. This is the first time however, that we have developed something that will see real world use. This places a whole new meaning on the work that is being done. The group works harder to ensure their client is happy and will be able to use the finished product. There was a world of difference between simulated projects that were completed in our coursework, and this final project.

Having gone through the entire system development lifecycle (SDLC) now, we have learned a great deal about each step along the way. We were quite ambitious with this project considering our team and timeframe, and that is reflected in our project scope and requirements. We have learned the consequences of promising a lot to a client, because the client then expects those features. Sometimes even the mention of a potential feature will heighten client expectations. We have learned to be more conservative and thoughtful about what we discuss with a client so as to not raise expectations to an unrealistic level.

Having set these ambitious goals, we found ourselves working extra hard to keep up with the work that we promised. This forced us to push deadlines and deploy later than expected. In many projects that we will undertake in our careers we may need to deal with a similar situation where we need to approach a client and explain to them why the project is being delayed. This is not an enjoyable experience, however it will prove to be something we will need to understand how to handle and we are grateful to learn to handle it early in our careers.

Another valuable experience that we had involved the division of work between the two of us; we had to take some time to sit down and see where each person's strengths and weaknesses were. Doing this involved being open about ourselves and the ability to be openly critical of each other while staying constructive. Division of labor based upon our experiences and strengths proved to be very helpful in finishing tasks in a timely manner and with the best

quality of work. These skills can be very useful in the future, whether it involves taking leadership in a project or interviewing for a position and explaining strengths and weaknesses.

Looking back at the many steps we took to get to this point is quite astonishing. While working on this project, one group member shared the project goals with some professional consultants who he worked with; all of whom said our timeline was tight and we would be hard pressed to accomplish our goals in the allotted time. We are both proud to say we have risen to the occasion. We feel that we have followed through on the SDLC and accomplished what we set out to accomplish, despite the challenges we faced along the way. We are both very proud of our work, and how we managed to collaborate effectively to accomplish our tasks within our allotted timeframe.

Works Cited

About Ruby. (2008). Retrieved September 7, 2008, from Ruby: http://www.ruby-lang.org/en/about/

Anderson, T. (2008, 31 July). *SQL Server 2008 Express: The Free Database from Microsoft*. Retrieved September 10, 2008, from DevX.com: http://www.devx.com/MicrosoftISV/Article/38728

Commonwealth of Massachusetts. (2008). *The Official Website of the Executive Office of Public Safety and Security (EOPSS)*. Retrieved September 9, 2008, from State Police Marine Section: http://www.mass.gov/?pageID=eopsterminal&L=4&L0=Home&L1=Homeland+Security+%26+Emergency+Response&L2=Response&L3=Emergency+Tactical+%26+Response+Teams&sid=Eeops&b=terminalcontent&f=msp_units_msp_marine_section&csid=Eeops

Commonwealth of Massachusetts. (2008). *The Official Website of the Executive Office of Public Safety and Security (EOPSS)*. Retrieved September 9, 2008, from State Police Underwater Recovery Team : http://www.mass.gov/?pageID=eopsterminal&L=4&L0=Home&L1=Homeland+Security+%26+Emergency+Response&L2=Response&L3=Emergency+Tactical+%26+Response+Teams&sid=Eeops&b=terminalcontent&f=msp_units_msp_underwater_recovery_page1&csid=Eeops

Dennis, A., Wixom, B. H., & Roth, R. M. (2006). *Systems Anaysis and Design.* USA: John Wiley & Sons, Inc.

Holovaty, A. ‚.-M. (2007, December 12). *Chapter 1: Introduction to Django*. Retrieved September 7, 2008, from The Django Book: http://www.djangobook.com/en/1.0/chapter01/

Lenz, P. (2008). *Simply Rails 2.* Collingwood VIC, Australia: Sitepoint Pty. Ltd.

Melo, S. (2008, September 3). Town of Dartmouth Harbormaster. (M. Fury Hopkins, & W. Price, Interviewers)

Microsoft. (2008). *Get Started: The Official Microsoft ASP .NET Site*. Retrieved September 7, 2008, from ASP.net: http://www.asp.net/get-started/

MySQL :: About MySQL. (2008). Retrieved September 10, 2008, from MySQL: http://www.mysql.com/about/

The PHP Group. (2008, September 6). *History of PHP and related projects - Manual*. Retrieved September 7, 2008, from PHP: http://us.php.net/manual/en/history.php

Town of Dartmouth. (2007). *2007 Town of Dartmouth Annual Report.* Retrieved September 2, 2008, from Town of Dartmouth - Annual Town Report: http://town.dartmouth.ma.us/Town%20Report%20Book%202007%20Final.pdf

Town of Dartmouth. (2001). *Dartmouth AR 2001.* Retrieved September 2, 2008, from Town of Dartmouth, MA - Official Home Page: http://www.town.dartmouth.ma.us/Dartmouth%20AR%202001.pdf

Town of Dartmouth. (n.d.). Mission Statement for Dartmouth Harbormaster Office and Waterways Department. Dartmouth, MA, United States Of America.

Town Of Dartmouth Waterways Management Department. (2005, May). Mission Statement. *Mission Statement for Dartmouth Harbormaster Office and Waterways Department* . Town Of Dartmouth, Massachusetts, United States of America.

United States Coast Guard. (2008, February 4). *USCG: About Us - Overview - Publication 1.* Retrieved September 4, 2008, from United States Coast Guard: http://www.uscg.mil/top/about/pub1.asp

Appendix I. Samples of Manual Documents

The following is an example list of manual documents used by the Town of Dartmouth Harbormaster's Office.

I.i Dinghy Renewal Form

Town of Dartmouth

WATERWAYS MANAGEMENT DEPARTMENT

STEVEN L. MELO JD. HARBORMASTER / WATERWAYS DIRECTOR

PO BOX 80218 DARTMOUTH, MA 02748

www.dartmouthharbormaster.com

FAX: (508) 990-7759

This Box for Collector Use

2008 Dinghy Renewal Form

«Person__FirstName» «Person__LastName»
«Person__Address»
«Person__City», «Person__State» «Person__Zip»

This form is used for the registration of a dinghy in the Town of Dartmouth, MA. A dinghy is defined as a vessel less than 12 feet in length.
Please print **ALL** changes in information *legibly* in blue or black ink.

2. Dinghy Location

2.a. Location (check one): ☒ Town Landing ☐ Hill St. ☐ Water St. ☐ Other: _____

2.b. Desired Storage Option (check one): ☒ Rack ($40) RACK # _____

*If you selected "**Rack**" in 2.b., you will be assigned a rack number by the Harbormaster Office. If you selected "**Shoreline**" in 2.b., storage is provided on a "first-come first-serve" basis in the designated area, with no guarantee of space or right to return to any space.*

DINGHY PERMIT is valid MARCH 15th through NOVEMBER 30th only. **Any dinghy not removed by November 30th shall be considered ABANDONED & REMOVED as refuse.**

3. Dinghy Information

3.a. Length: _____ 3.b. Color: _____

3.c. Manufacturer: _____ 3.d. Model: _____

3.e. HIN: _____

3.f. Type (check one): ☐ Fiberglass ☐ RIB ☐ Soft-bottom Inflatable ☐ Other: _____

3.g. Registration*: _____ 3.h. Registration State: _____

3.i. Identifying Marks/Features:

By signing this form, you affirm that you understand your obligations as a boater in Dartmouth and that the Town of Dartmouth is not responsible in any way for any damage, theft, abuse, or other harm that may befall your property while stored on Town land. It is an owner's responsibility to properly secure his or her vessel.

State law requires the registration of **any boat that is powered by a motor and operated on public waterways in Massachusetts. Registration is required even if the motor is not the primary means of propulsion for that boat.*

Signature: _____ Date: _____

I.ii Waterways Use Application: Form A (BOAT PERMIT)

Town of Dartmouth
WATERWAYS MANAGEMENT DEPARTMENT
STEVEN L. MELO, J.D. HARBORMASTER/WATERWAYS DIRECTOR
WWW.DARTMOUTHHARBORMASTER.COM
FAX: (508) 990-7759

WATERWAYS USE APPLICATION
2008 - Form A (BOAT PERMIT)

Boat is kept on: Mooring #: _____ **Dock Location/Slip #:** _____

-Boat Information- Please *Complete* All Information

Boat Name: _____

Registration/Doc. # (REQUIRED): _____ Primary Hull Color: _____

Boat Length: _____ *Boat* Draft: _____

Boat Type: ☐ Power ☐ Sail *Boat* Manufacturer: _____ Year Built: _____

Boat Owner Name: _____

Boat Owner Mailing Address: _____

City / Town: _____

State: _____ Zip Code: _____

Boat Owner Home Tel.: _____ *Boat Owner* Emergency Tel.: _____

***Boat Owner* E-mail Address**: _____

Boat Owner Seasonal Address (dates applicable): _____

* E-mail addresses will be kept private. It is very important that you provide it for our new notification services.

-Mooring Information-

Mooring Regulations require that Recreational Mooring Permittee must be **SAME** as Boat Owner.

If renting a Commercial Mooring, list Rental Company Name: _____

⇒ **I affirm that I have read all of the applicable Waterways regulations and I understand my obligations as a Dartmouth Waterways user.** (*See all regulations at:* **www.dartmouthharbormaster.com**)
⇒ ***Any false statements are grounds for forfeiture of mooring locations.***
⇒ **Form and Fee must be mailed to address on reverse side.**

Signature: _____ Date: _____

(See reverse side)

I.iii Renewal Form

Town of Dartmouth
Collector - Waterways Management
PO Box 80218
Dartmouth, MA 02748
www.dartmouthharbormaster.com

2008 RENEWAL FORM

Please make any corrections to your **contact** information below on this form.

We have your contact information as:

Phone: _____

Alt Phone: _____

*E-mail: _____

* e-mail is kept private and helps us keep you informed

COMPLETED RENEWAL APPLICATIONS MUST BE RECEIVED AT THE TOWN COLLECTOR BY 4:00 p.m. MONDAY MARCH 31, 2008. Mooring registration forms/fees not RECEIVED by March 31, 2008 are assessed an additional $50.00 LATE PENALTY per town by-law. No Mooring Permit may be renewed after May 1, 2008.

- Mooring permits with *overdue* or *lapsed* inspection cannot be renewed. It is the sole responsibility of each mooring permit holder to provide timely mooring inspection reports.
- NO RECREATIONAL MOORING MAY BE RENTED – Violations cause mooring permit revocation.
- Change in "Vessel of Record" assigned to a mooring permit requires prior written Harbormaster approval.

Vessel of Record Name	Length Overall (LOA)	Boat State/Federal Registration No:	Dock or Mooring Number	Last Filed Mooring Inspection	Next Inspection Due:	Boat + Mooring fee
TOTAL AMOUNT DUE by 3/31/08 (payable to "Town of Dartmouth") ⇨						

-Annual *VESSEL* Waterways Use Fee Schedule-	
Resident	***Non-Resident***
$2.60 per foot x Actual vessel Length Overall* **(LOA)** * round up LOA to whole foot only	**$6.20 per foot x Actual** vessel Length Overall* **(LOA)** * round up LOA to whole foot only
A Resident is one of the following: (**Proof** of the following must be furnished upon request) A person who ***pays real estate taxes*** to the Town of Dartmouth, is domiciled in the Town of Dartmouth, ***registered voter*** in the Town of Dartmouth, A ***spouse or dependent*** of any of the above	
-Annual *MOORING* Fee Permits for Residents and Non-Residents $30.00 each included in total amount above-	

READ:

⇒ I affirm that I have read all applicable Waterways regulations and I understand my obligations as a Dartmouth Waterways user. (Available upon request & at www.dartmouthharbormaster.com)

⇒ Any false statements are grounds for forfeiture of mooring locations.

⇒ FORM CANNOT BE PROCESSED WITHOUT SIGNATURE ✍.

Signature: _____ ⇦(Required) Date: _____

MAIL THIS FORM AND FEES TO: (checks payable to "*Town of Dartmouth*")
Collector - Dartmouth Waterways; P.O. Box 80218; Dartmouth, MA 02748
Payments made in person are **ONLY** accepted at the **Collector's Office**, (2nd floor of Dartmouth Town Hall).
$50 LATE FEE APPLIES IF NOT RECIEVED BY 3/31/08
Mooring sites not renewed by 5/1/08 are re-assigned to the waiting list

Appendix II. Technical Documentation

This appendix includes instructions on the installation of the Ruby on Rails language pack , the suggested Integrated Development Environment (IDE) , mySQL and a suggested reading list.

Dartmouth Harbormaster System

Technical Documentation

William D. Price

January 2009

Installing

Before you can run this software on a server, you must install the following items:

- Ruby 1.8.6
- MySQL 5.0
- Rubygems
- Rails 2.1.2
- Configuration gems

Ruby 1.8.6 & Rubygems

Ruby can be installed to a Windows based machine by downloading and executing the file ruby186-26.exe from http://rubyforge.org/frs/?group_id=167.

It is a standard Windows installer; you can use all standard options. Select the checkbox to enable Rubygems as well.

MySQL

Installing MySQL involves downloading a version of MySQL 5.0 from http://dev.mysql.com/downloads/.

Rails 2.1.2

Open a command prompt and enter the following line:

gem install rails –v 2.1.2

Some text will scroll through the command prompt, be patient and wait for a new line to appear.

Configuration Gems

Similar to the install of rails, a few more gems are needed in order to run the application to its full capability. Run the follow commands form a command prompt:

gem install prawn –include-dependencies

gem install mongrel –include-dependencies

gem install mysql

Optional Installations:

If you wish to edit any of the code, the recommended Integrated Development Environment (IDE) is NetBeans. You can download NetBeans at http://www.netbeans.org/downloads/. Select the Ruby version. During the install, uncheck the box for GlassFish.

Configuring the Database

Open MySQL by typing into the command prompt (replacing username with your username created during installation):

mysql –u username –p

Once you are logged in (the command prompt line will read "mysql>") enter the following commands:

create database harbor_dev

create database harbor_test

create database harbor_prd

These commands create three databases for use in each of the rails environments, development, testing, and production.

Configuring Rails to communicate with MySQL

Open the Rails application in NetBeans and open the folder called Configuration. Open the file in this folder called database.yml. In this file, enter the username and password you set up for MySQL for each environment. Save the file, and then let's test to make sure the communication works.

In NetBeans, right click on the main project folder (HarborSystem) and select the option: Migrate Database => To Current Version.

A database migration automatically creates all the tables in MySQL. If there are no errors, it is set-up correctly.

You can then click on the Run button n the top of NetBeans, and it should open a web browser with a development version of the web-page.

Loading Data

To load data into the system, CSV fields should be generated from the old data, and then formatted into the correct column order for the corresponding table. The order of columns and data types can be obtained from running the SQL command: *describe tablename*.

To load the data from the CSV into the table, please reference the MySQL documentation page for LOAD DATA located at:

http://dev.mysql.com/doc/refman/5.0/en/load-data.html

Manipulating Data

All data stored in the system can be manipulated within the application on the web by clicking Edit, however if something needs to be changed in the back-end, the data can be manipulated by using standard SQL queries when logged into the correct database.

**NOTE: Be careful when editing related data. The database does not cascade, so deletion of an object that has related data can cause errors!*

How Rails Operates

There are three core types of files that work together to create a Rails web application. Each of these files handles a different aspect of the application. The files are:

- Models
- Controllers
- Views

Models – These files are located within the Model folder in the project directory. These files define relationships between different entities, and also define special attributes for a data type. For example, in the Person model, there is a function that will return the person's first name concatenated to their last name. This can be called like any other attribute that exists within the table.

Controllers - These files are located in the Controller folder in the project directory. Functions are contained in these files for each webpage that exists for a given model. Different variables that are defined here can be called in the associated View code. These files are important to keep the database searching and calculations on the server end, to minimize the amount of work that the client browser needs to do. A good website wants as much calculation done on the server end as possible to enhance performance. A special Application controller also exists with functions that can be called on any View page.

Views – These files are embedded ruby HTML files (view.html.erb) that are read by the browser and create the webpage that the end user will see. They are located in the View folder of the application directory, and are broken down into views for each model. If any changes need to be made to a webpage that is displayed, the view files are the place to make edits. The file is written with standard HTML tags, with special tags surrounding ruby code that allows it to be executed. The format for embedded ruby code is as follows: <% code %>. These HTML files can also follow CSS stylesheets, which are located in the root/public/stylesheets folder in the application directory.

For more information on how Rails works, please see the recommended articles at the end of this section.

Deploying the system on Apache 2.2

In order to deploy this system on Apache 2.2 (assumed installed to c:/apache), some new gems must be installed. First, run the command:

gem install mongrel_service --include-dependencies

After this installation is complete, install mongrel as a windows service on the server by navigating to the application directory in a command prompt and using the following commands:

mongrel_rails service::install –N HarborSystem_4001 –p 4001 –e production

mongrel_rails service::install –N HarborSystem_4002 –p 4002 –e production

Now mongrel will run two instances of the application automatically on server startup. Having two mongrel instances helps with application load balancing.

Now navigate to the apache/conf and open the file *httpd.conf*. Add the following line to the bottom of the file:

Include httpd-proxy.conf

Save and close the file, and then create a new file in the apache/conf folder called: *httpd-proxy.conf*. Enter the code included in **Appendix III. httpd-proxy_code** in this file and save.

Restart the Apache server and navigate to the domain specified during your install of Apache and you should see your application!

Plug-ins

This application utilizes two plug-ins that allow it to expand functionality. These files are located in the vender/plug-ins folder in the application directory. Included is the name of the plug-in, and website(s) with useful resources.

Restful_authentication – For creating unique user sessions.

http://wiki.rubyonrails.org/rails/pages/restful_authentication

Prawnto – For creating PDF files.

http://www.cracklabs.com/prawnto

Recommended Reading

To gain a better understanding of this software, and the platform it was developed on, the following articles and tutorials are recommended for reading and for practice.

Ruby on Rails Complete API (For reference)

http://api.rubyonrails.org/

Learn Ruby on Rails: The Ultimate Beginner's Tutorial

http://www.sitepoint.com/article/learn-ruby-on-rails/

Deploying Rails Applications: A Step-By-Step Guide

http://www.scribd.com/doc/7281218/

Appendix III. httpd-proxy_code

This appendix contains the source code found in the httpd-proxy.conf document that is located on the Harbormaster server in C:\apache\conf. The code specifies a configuration in which all requests coming through the apache server are routed to two instances of Mongrel web server. For more information on the use of this code, please see the deployment section of Appendix II. Technical Documentation.

```
<VirtualHost *:80>
      ServerName dartmouthharbormaster.webhop.net
      DocumentRoot c:/web/HarborSystem/public

      <Directory "c:/web/HarborSystem/public">
        Options FollowSymLinks
        AllowOverride None
        Order allow,deny
        Allow from all
      </Directory>

      #Configure Mongrel Instance
      <Proxy balancer://mongrel_cluster>
        BalancerMember http://127.0.0.1:4001
        BalancerMember http://127.0.0.1:4002
      </proxy>

      RewriteEngine On

      # Uncomment for rewrite debugging
      #RewriteLog logs/your_app_deflate_log deflate
      #Rewrite LogLevel

      # Check for maintenance file and redirect all system requests
      RewriteCond %{DOCUMENT_ROOT}/system/maintenance.html -f
      RewriteCond %{SCRIPT_FILENAME} !maintenance.html
      RewriteRule ^.$ /system/maintenance.html [L]

      #Rewrite index to check for static
      RewriteRule ^/$ /index.html [QSA]

      # Rewrite to check for Rails cached page
      RewriteRule ^([^.]+)$ $1.html [QSA]

      # Redirect all non-static requests to cluster
      RewriteCond %{DOCUMENT_ROOT}/%{REQUEST_FILENAME} !-f
      RewriteRule ^/(.*)$ balancer://mongrel_cluster%{REQUEST_URI} [P,QSA,L]

      # Deflate
      AddOutputFilterByType DEFLATE text/html text/plain text/xml
      BrowserMatch ^Mozilla/4 gzip-only-text/html
      BrowserMatch ^Mozilla/4\.0[678] no-gzip
      BrowserMatch \bMSIE !no-gzip !gzip-only-text/html

      # Uncomment for deflate debugging
      #DeflateFilterNote Input input_info
      #DeflateFilterNote Output output_info
      #DeflateFilterNote Ratio ratio_info
      #LogFormat '"%r" %{output_info}n/%{input_info}n (%{ratio_info}n%%);
deflate
      #CustomLog logs/your_app_deflate_log deflate
      ErrorLog logs/your_app_error_log
      CustomLog logs/your_access_log combined
</VirtualHost>
```

Appendix IV. User Manual

This includes instructions for the user on how to add and manipulate data in the system. Also included is how to add billing rates for the year and produce bills.

Dartmouth Harbormaster System

User Manual

William D. Price & Millisent Fury Hopkins

January 2009

Contents

Overview 104

Explanation of Page Elements 105

Accessing the system 106

Navigating the system 107

Entering Data 108

Viewing data 110

Searching Data 111

Nuances of the System 112

- **Entering a New Mooring & Waitlist** 112
- **Marking a Ticket as Paid** 113

Entering Billing Rate Information 115

- **Generating Bills** 116
- **Record a Payment for a Bill** 119

Conclusion 121

Overview

The Dartmouth Harbormaster System is a web-based application that was developed for use by the Harbormaster and Assistant Harbormasters for tracking data important to their function. The following pages outline the use of this system.

We begin with a brief explanation of page elements to ensure you understand the different methods we will reference to enter data and navigate the system. We then discuss creating a user account, and finding your way to different types of data within the system. From here you will learn how to enter data, and then view whatever you have entered.

The system contains a strong live search function, which is explained the next section. We follow with a section that explains some of the nuances of the system, and we end with an explanation of how to generate invoices. Please follow these instructions whenever using the system to ensure the system works correctly for you for many years to come.

Explanation of Page Elements

Below is a listing of elements that are used on the web pages in this system.

- Drop Down Menu –

    - These menus are filled with a list of selectable values. Click the arrow on the right side of the menu and the list expands. Then select the item on the list you wish to use.
- Text Field -

    - This field is a blank box that allows you to enter a value into it. It is recommended to write "N/A" into a text field that you have no data for instead of leaving it blank. The example above has a value entered already.
- Button -

    - Clicking on a button performs an action in the system. In the example above clicking the button would generate all the invoices for the current year. For more information on this particular button see page 15.
- Link -

    - A link is a piece of text that is underlined, and either blue or purple. Clicking a link navigates you to a new webpage. The example above would navigate you to a page to create a new invoice.

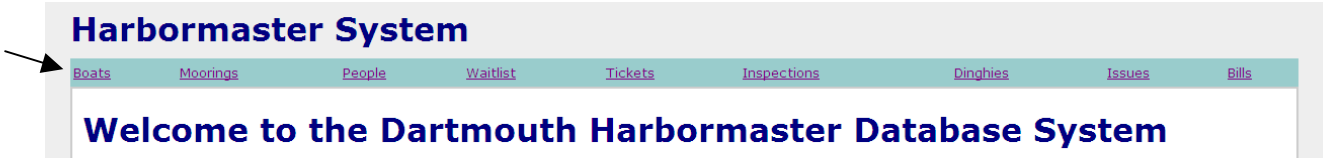
Accessing the system

Before you can work with the system you must be in the system as a "user". To become a "user" have someone who is already a user on the system log on and access http://dartmouthharbormaster.webhop.net/signup. The system comes with an administrator account already created. Please see your system administrator for more information.

Choose a login name, fill in the email address field and select a password. Once "Sign up" is clicked and the login is unique, a new user has been created. You may now access the system via http://dartmouthharbormaster.webhop.net/.

Navigating the system

Users are able to access different portions of the system by clicking on the links along the top of the page. Clicking on a link will bring you to the Listing page for that type of data. From the listing page the user is able to look at and edit a specific entry in the system. For example, clicking the “Boat” link will navigate you to the page that contains a listing of all boats entered into the system.

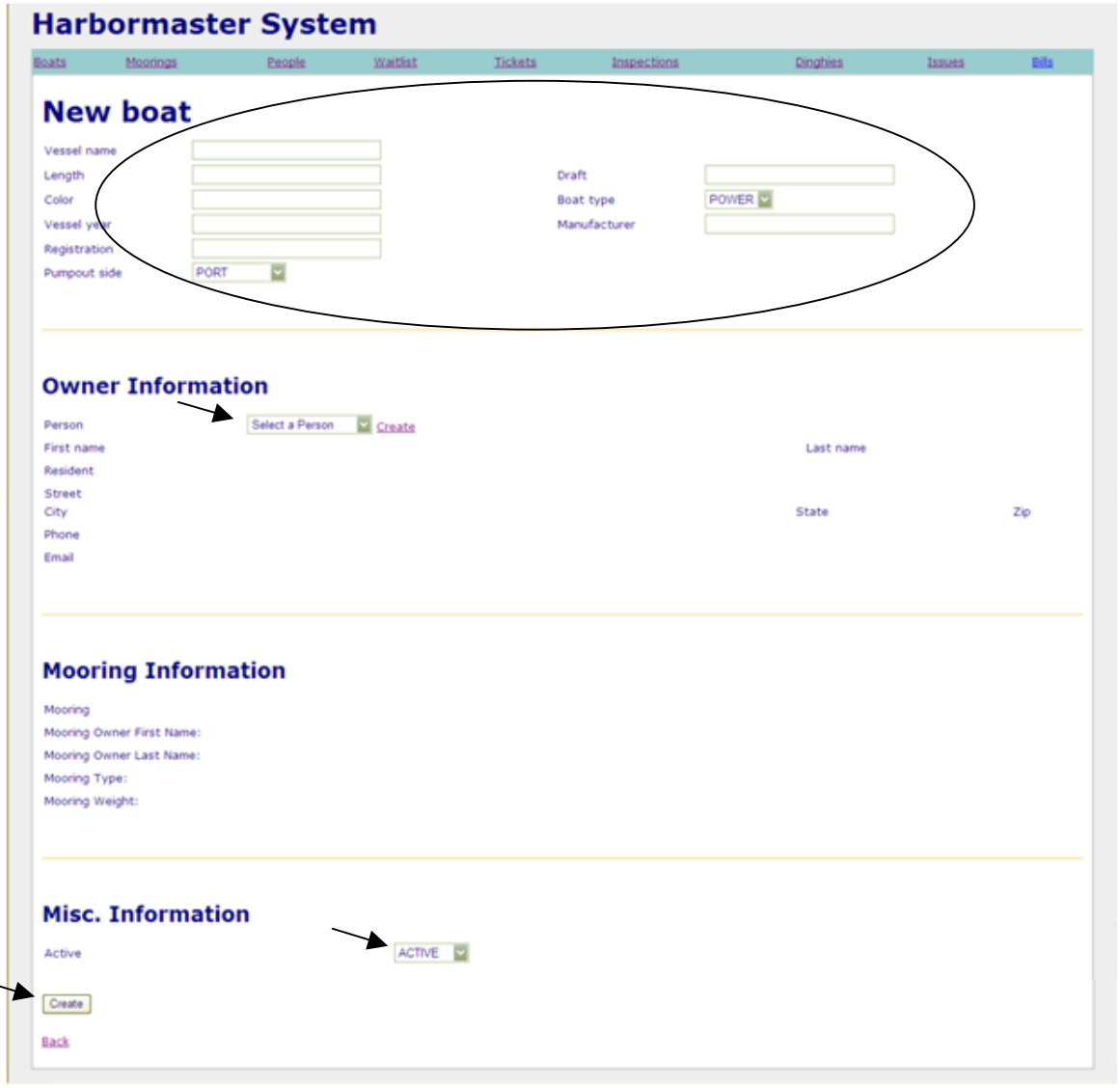


Entering Data

To input a new entry into the system, click on the "New" link at the bottom of the Listing page. For example, when creating a new boat, you navigate to the boats area of the system by clicking the "Boats" link at the top of the page. Then scroll to the bottom of the list page and click the link that reads "New Boat". See the diagram below for more details.

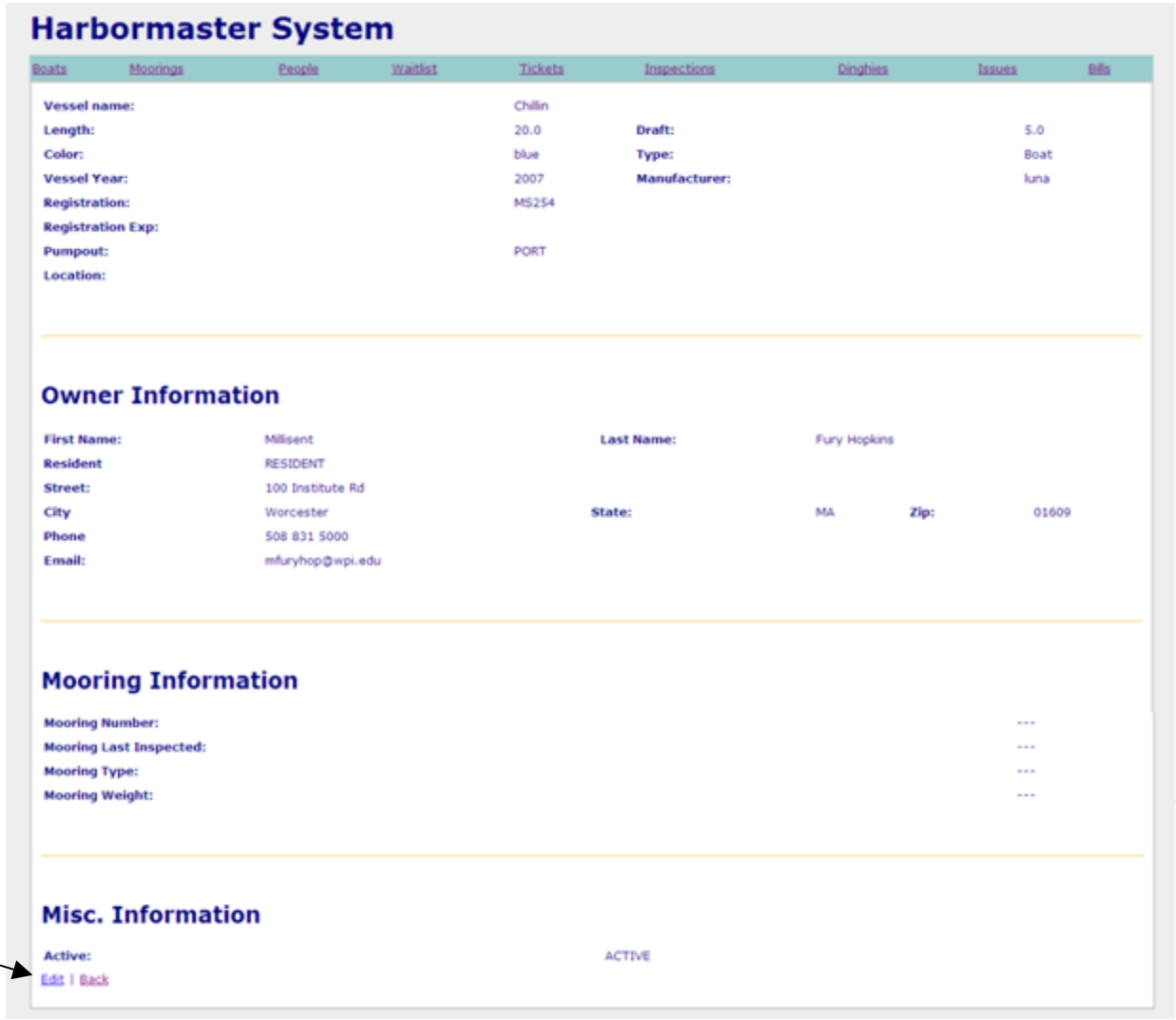
This will bring you to a blank entry page for you to input the information for a new item in the system. Fill in the applicable fields, select the necessary drop downs and click the "Create" button to create the new entry in the system.

In the example on the next page you see the "New Boat" page. The circled area contains text fields and drop down menus that are labeled for the type of data they contain. Enter the name of the boat into "Vessel Name", the draft into "Draft" and so on until you enter the required information. For an explanation on drop down menus please see page 4. In the next section, select the owner of this boat from the drop down menu labeled as "Person". This drop down will contain a list of each person entered into the system already. Be sure to click the correct person, because this will link the boat to that person. At the bottom you select from one last drop down if the boat if active or not.


Harbormaster System
Boats
Moorings
People
Waitlist
Tickets
Inspections
Dinghies
Issues
Bills
New boat
Vessel name
Length
Color
Vessel year
Registration
Pumpout side
PORT
Draft
Boat type
POWER
Manufacturer
Owner Information
Person
Select a Person
Create
First name
Last name
Resident
Street
City
State
Zip
Phone
Email
Mooring Information
Mooring
Mooring Owner First Name:
Mooring Owner Last Name:
Mooring Type:
Mooring Weight:
Misc. Information
Active
ACTIVE
Create
Back

Viewing data

Once the item is created you will be brought to the show page where you can check over the entry for any mistakes. If there are, simply click “Edit” and make the appropriate changes. Then click the “Update” button to save the changes.



By clicking “Back” you will return to the Listing page to find your new entry is now listed.

Searching Data

At the top of each Listing page is a “Search By” drop down menu and a text field. To do a search of the items on the listing page simply choose the type of field you would like to search by and type into the textbox what you are searching for. In the example below the user is searching for all boats owned by people with the last name of “Price”. This is accomplished by selecting OWNER_LAST_NAME” from the drop down menu, and then entering the word “Price” into the text field below it. Search results will be listed automatically as you type.

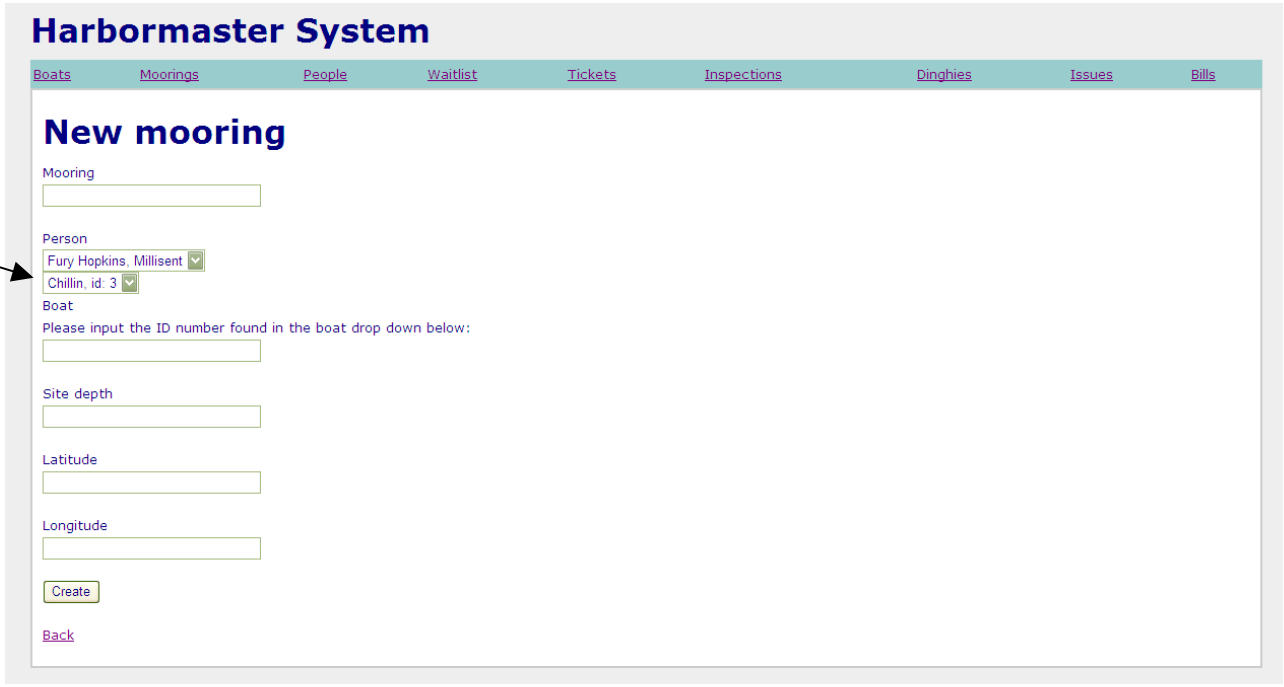
Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Search By: OWNER_LAST_NAME
Price
'Price' found 2 time(s)!
2008 Guiding Light POWER 30.0 White Price, William 1A-1A Show | Edit
2008 Waiting Boat POWER 15.0 White Price, William --- Show | Edit
Listing boats
Year Name Type Length Color Owner Name Mooring Number
2008 Guiding Light POWER 30.0 White Price, William 1A-1A Show | Edit
2008 Waiting Boat POWER 15.0 White Price, William --- Show | Edit
2007 Chillin SAIL 20.0 blue Fury Hopkins, Millisent --- Show | Edit
2007 Boat Doe POWER 16.0 White Doe, John 1B-1B Show | Edit
New boat

Nuances of the System

Using the system follows most of the basic procedures listed above generally. However there are some situations that require special instructions. This section explains those situations, and the appropriate ways to handle them.

Entering a New Mooring & Waitlist

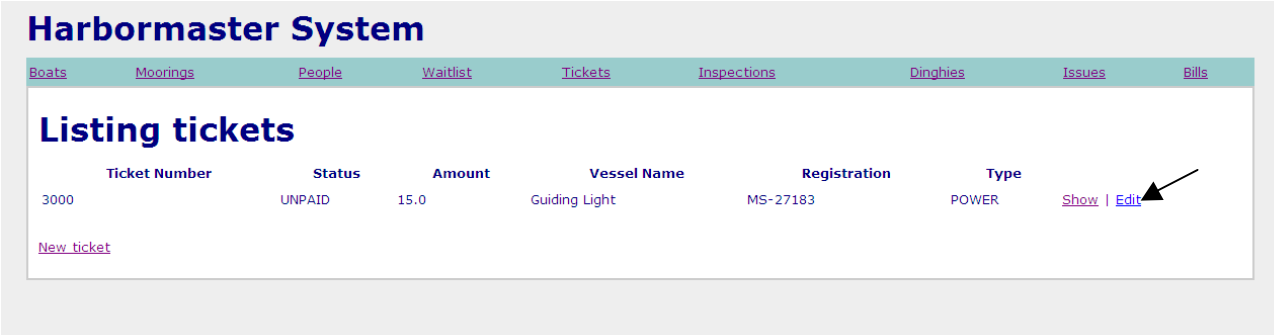
When adding a new mooring or waitlist, it is necessary to not only select a boat from the drop down list but also enter the id number exactly as listed in the following textbox. The id is provided to you listed next to the boat name. In the example below, the user has selected a person from the first drop down menu, and a boat from the second drop down menu. You can see in boat drop down menu the listed entry is "Chillin, id: 3". The user must enter the value "3" into the text field below where indicated. Be extra careful to type the correct value as an error could link the wrong boat to this mooring entry. The waitlist page is set up the same as the mooring page and these instructions can be applied to both.



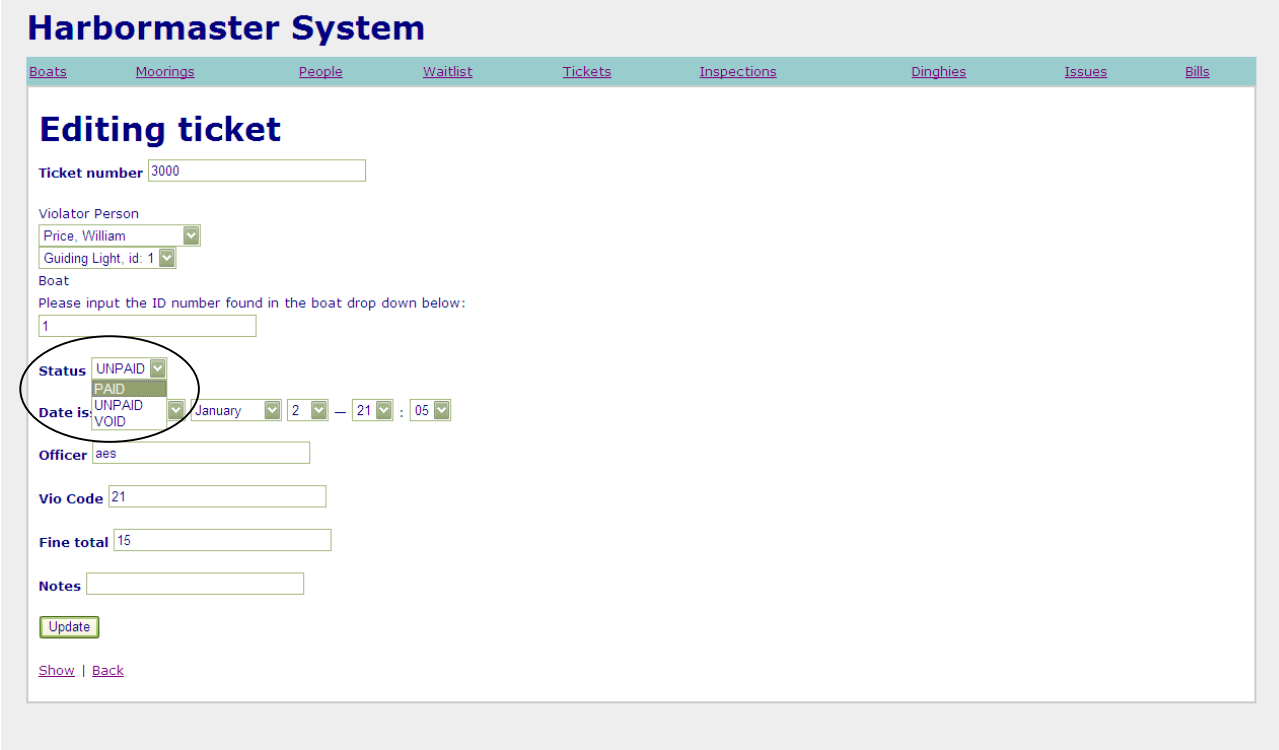
Marking a Ticket as Paid

To mark a Ticket as Paid, click on the "Tickets" link at the top of the page.

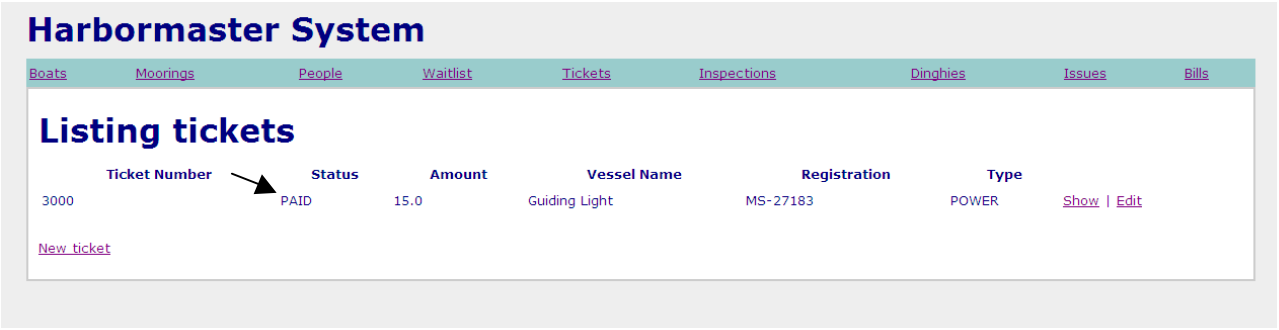
This will bring you to the Tickets Listing page. Click the "Edit" link next to the bill you would like to mark as paid.



Change the “Status” drop down menu to “PAID” and click the “Update” button to save your changes.



On the Tickets Listing page you will see that the ticket is now listed as paid.

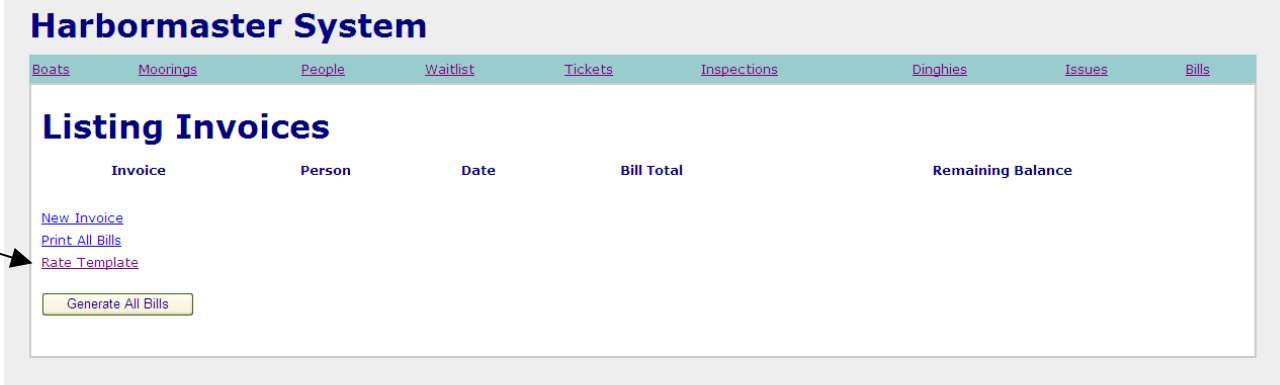

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Listing tickets
Ticket Number Status Amount Vessel Name Registration Type
3000 PAID 15.0 Guiding Light MS-27183 POWER Show | Edit
New ticket

Entering Billing Rate Information

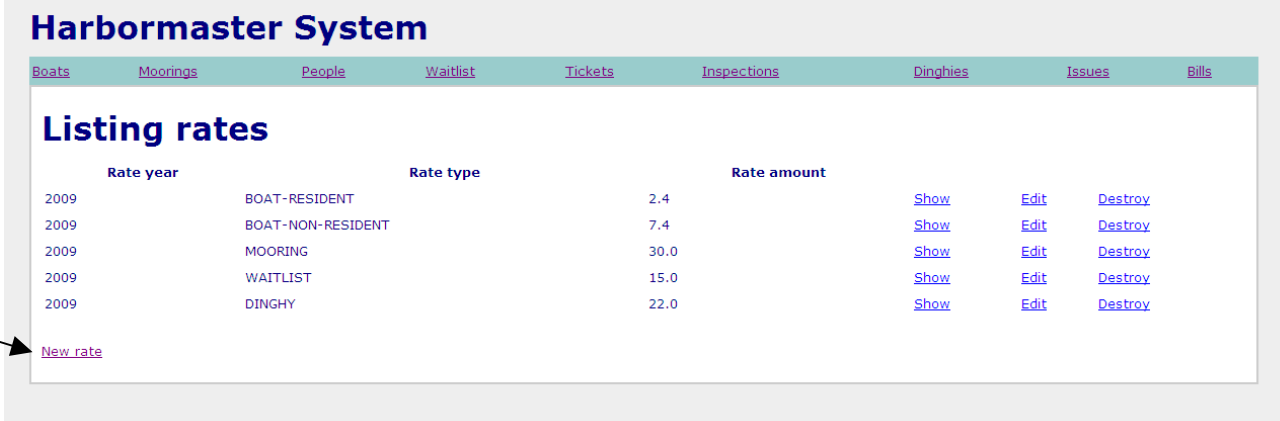
The system creates bills for each person in the database based on billing rates for the current year. To enter billing rate information, navigate to the "Bills" page by clicking the link at the top right of the page.

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Welcome to the Dartmouth Harbormaster Database System

On the Bills Listing page click on the "rate template" link at the bottom of the page.



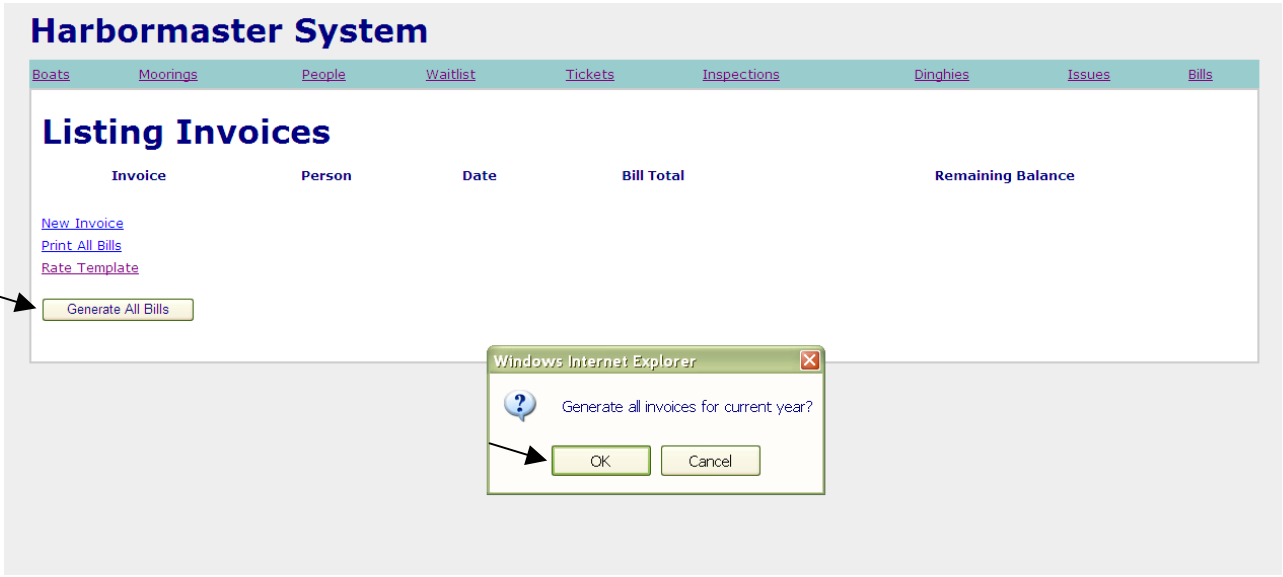
You are brought to the Rate Listing page, here you can add, edit or delete a rate. To add a new rate, click on the "New rate" link at the bottom left of the page.


Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Listing rates
Rate year Rate type Rate amount
2009 BOAT-RESIDENT 2.4 Show Edit Destroy
2009 BOAT-NON-RESIDENT 7.4 Show Edit Destroy
2009 MOORING 30.0 Show Edit Destroy
2009 WAITLIST 15.0 Show Edit Destroy
2009 DINGHY 22.0 Show Edit Destroy
New rate

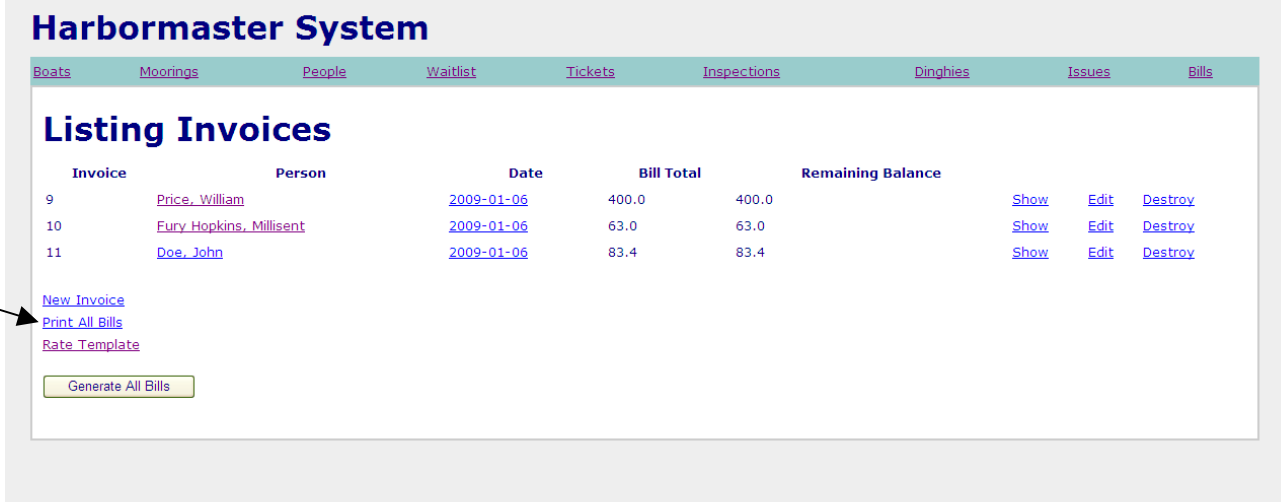
Fill in all the fields and click the “Create” button to create a new rate. This will bring you back to the Rate Listing page.

Generating Bills

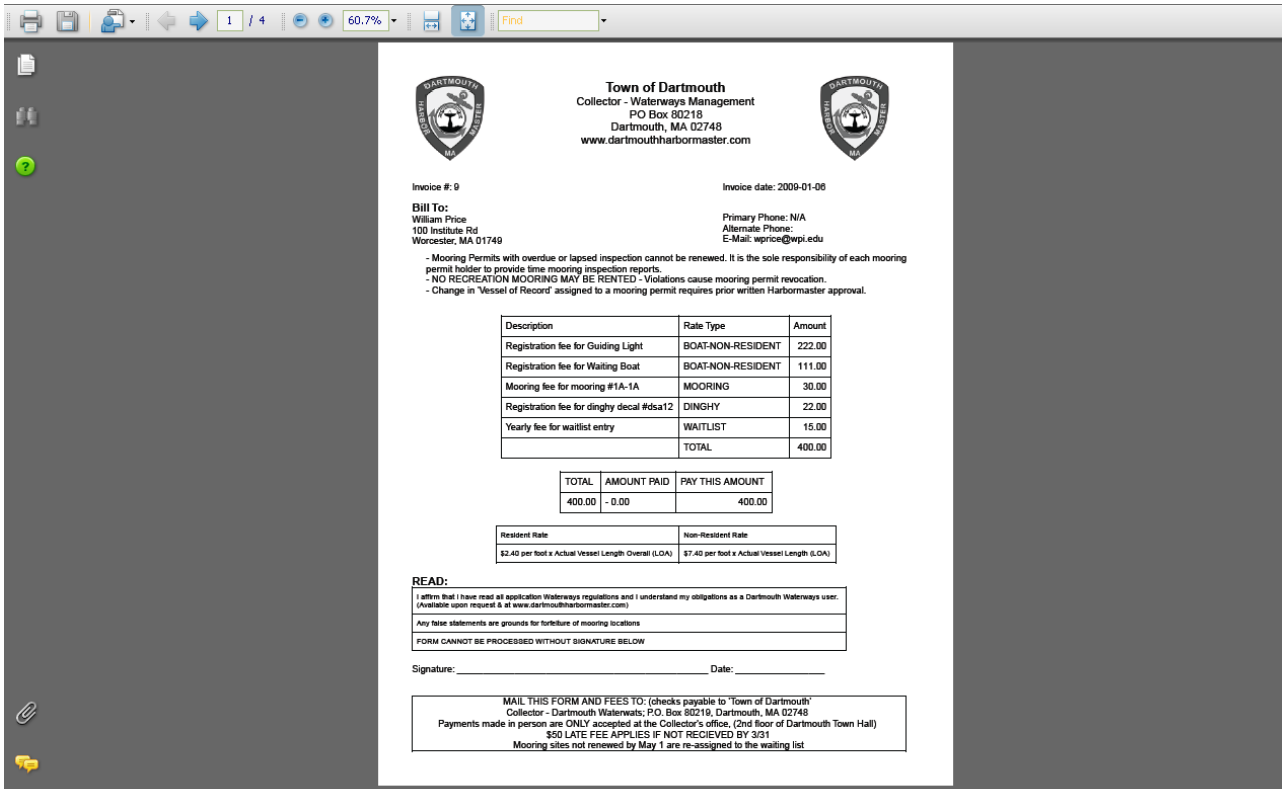
To generate bills for the current year, click on the Bills link at the top right of the page and click the “Generate all Bills” button. You will receive a confirmation pop-up, click “OK” and the invoices for the current year are now listed. Be patient as this may take some time.



To print all the bills for the current year click the “Print All Bills” link on the bottom right of the page.


Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Listing Invoices
Invoice Person Date Bill Total Remaining Balance
9 Price, William 2009-01-06 400.0 400.0 Show Edit Destroy
10 Fury Hopkins, Millisent 2009-01-06 63.0 63.0 Show Edit Destroy
11 Doe, John 2009-01-06 83.4 83.4 Show Edit Destroy
New Invoice
Print All Bills
Rate Template
Generate All Bills

A PDF will open in a new widow with all the bills for the year from the system in it, ready to print.



Town of Dartmouth
Collector - Waterways Management
PO Box 80218
Dartmouth, MA 02748
www.dartmouthharbormaster.com

Invoice #: 9
Invoice date: 2009-01-06

Bill To:
William Price
100 Institute Rd
Worcester, MA 01749

Primary Phone: N/A
Alternate Phone:
E-Mail: wprice@wpi.edu

- Mooring Permits with overdue or lapsed inspection cannot be renewed. It is the sole responsibility of each mooring permit holder to provide time mooring inspection reports.
- NO RECREATION MOORING MAY BE RENTED - Violations cause mooring permit revocation.
- Change in 'Vessel of Record' assigned to a mooring permit requires prior written Harbormaster approval.

Description	Rate Type	Amount
Registration fee for Guiding Light	BOAT-NON-RESIDENT	222.00
Registration fee for Waiting Boat	BOAT-NON-RESIDENT	111.00
Mooring fee for mooring #1A-1A	MOORING	30.00
Registration fee for dinghy decal #dsa12	DINGHY	22.00
Yearly fee for waitlist entry	WAITLIST	15.00
	TOTAL	400.00

TOTAL	AMOUNT PAID	PAY THIS AMOUNT
400.00	- 0.00	400.00

Resident Rate	Non-Resident Rate
$2.40 per foot x Actual Vessel Length Overall (LOA)	$7.40 per foot x Actual Vessel Length (LOA)

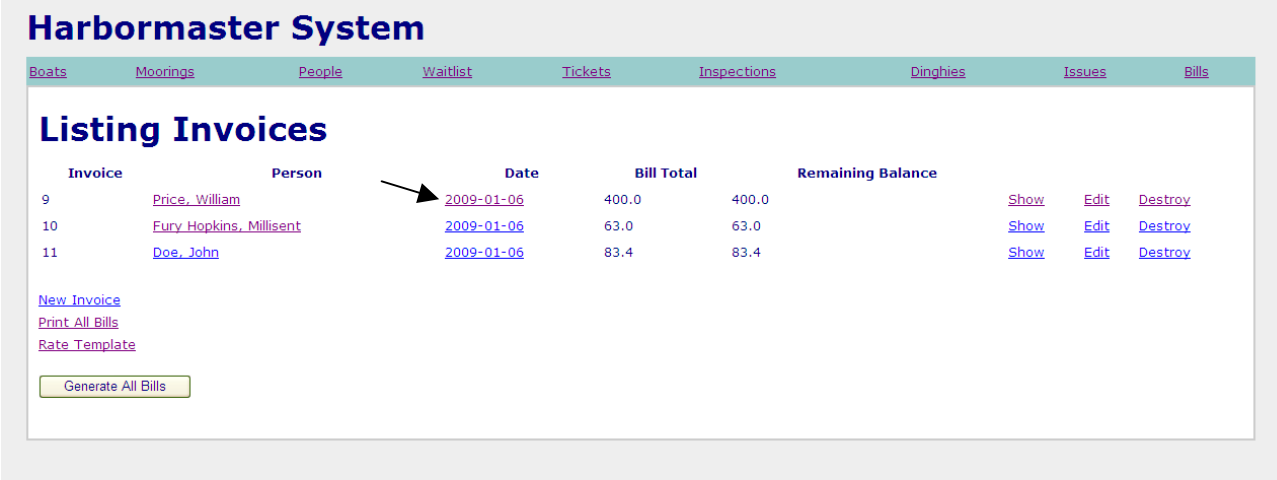
READ:

I affirm that I have read all application Waterways regulations and I understand my obligations as a Dartmouth Waterways user. (Available upon request & at www.dartmouthharbormaster.com)
Any false statements are grounds for forfeiture of mooring locations
FORM CANNOT BE PROCESSED WITHOUT SIGNATURE BELOW

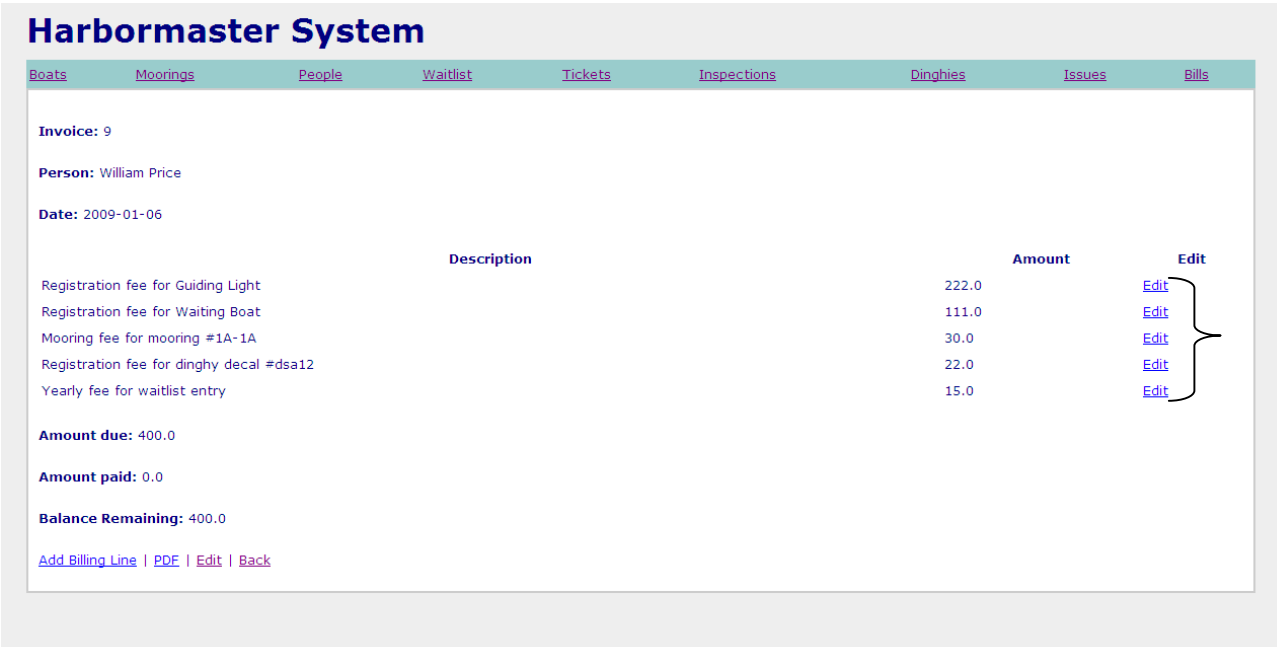
Signature: _____ Date: _____

MAIL THIS FORM AND FEES TO: (checks payable to "Town of Dartmouth"
Collector - Dartmouth Waterwats; P.O. Box 80219, Dartmouth, MA 02748
Payments made in person are ONLY accepted at the Collector's office, (2nd floor of Dartmouth Town Hall)
$50 LATE FEE APPLIES IF NOT RECIEVED BY 3/31
Mooring sites not renewed by May 1 are re-assigned to the waiting list

If for some reason an individual bill is incorrect, you may return to the system by clicking “Back” on your web browser or via http://dartmouthharbormaster.webhop.net/ and then to the Bills Listing page by clicking the “Bills” link at the top right of the page. Click on the Invoice date of the bill you would like to edit.

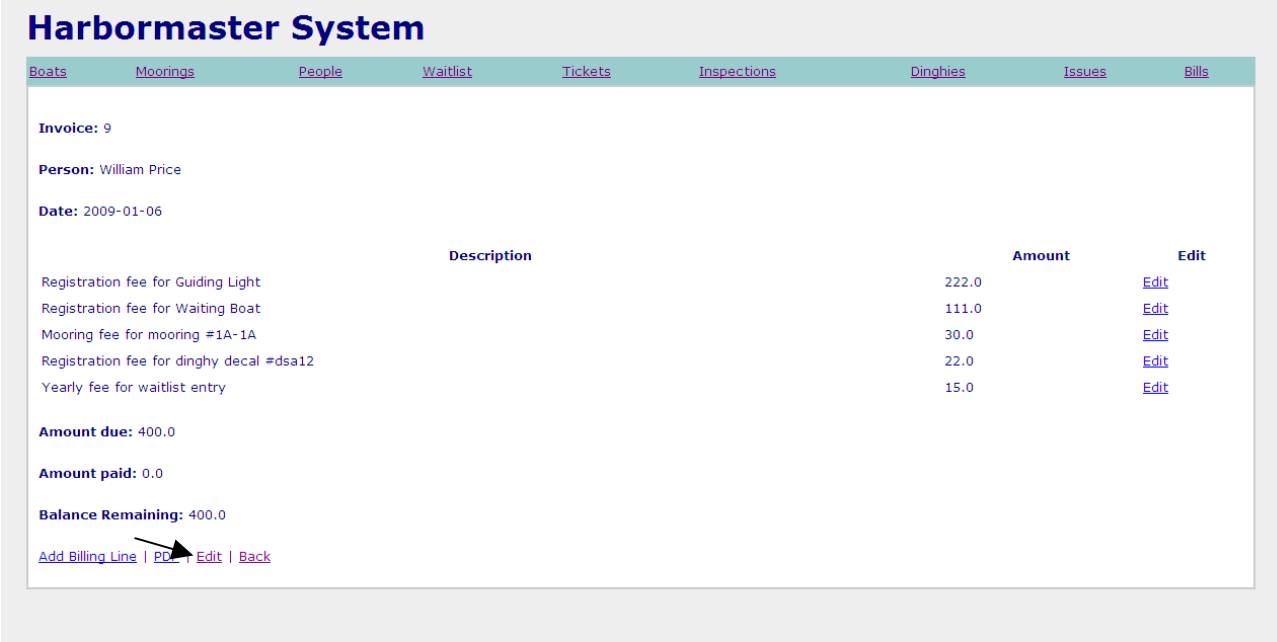

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Listing Invoices
Invoice Person Date Bill Total Remaining Balance
9 Price, William 2009-01-06 400.0 400.0 Show Edit Destroy
10 Fury Hopkins, Millisent 2009-01-06 63.0 63.0 Show Edit Destroy
11 Doe, John 2009-01-06 83.4 83.4 Show Edit Destroy
New Invoice
Print All Bills
Rate Template
Generate All Bills

Once on the invoice page you would like to edit, you may edit a specific billing line by clicking the "Edit" link to the right of the line item.

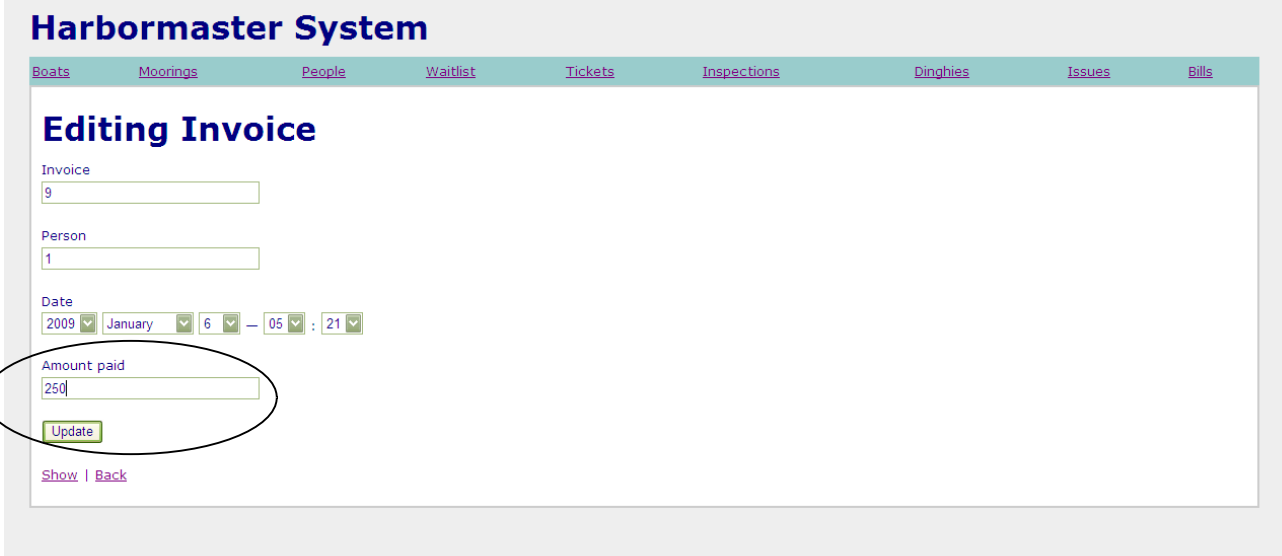

Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Invoice: 9
Person: William Price
Date: 2009-01-06
Description Amount Edit
Registration fee for Guiding Light 222.0 Edit
Registration fee for Waiting Boat 111.0 Edit
Mooring fee for mooring #1A-1A 30.0 Edit
Registration fee for dinghy decal #dsa12 22.0 Edit
Yearly fee for waitlist entry 15.0 Edit
Amount due: 400.0
Amount paid: 0.0
Balance Remaining: 400.0
Add Billing Line | PDF | Edit | Back

Record a Payment for a Bill

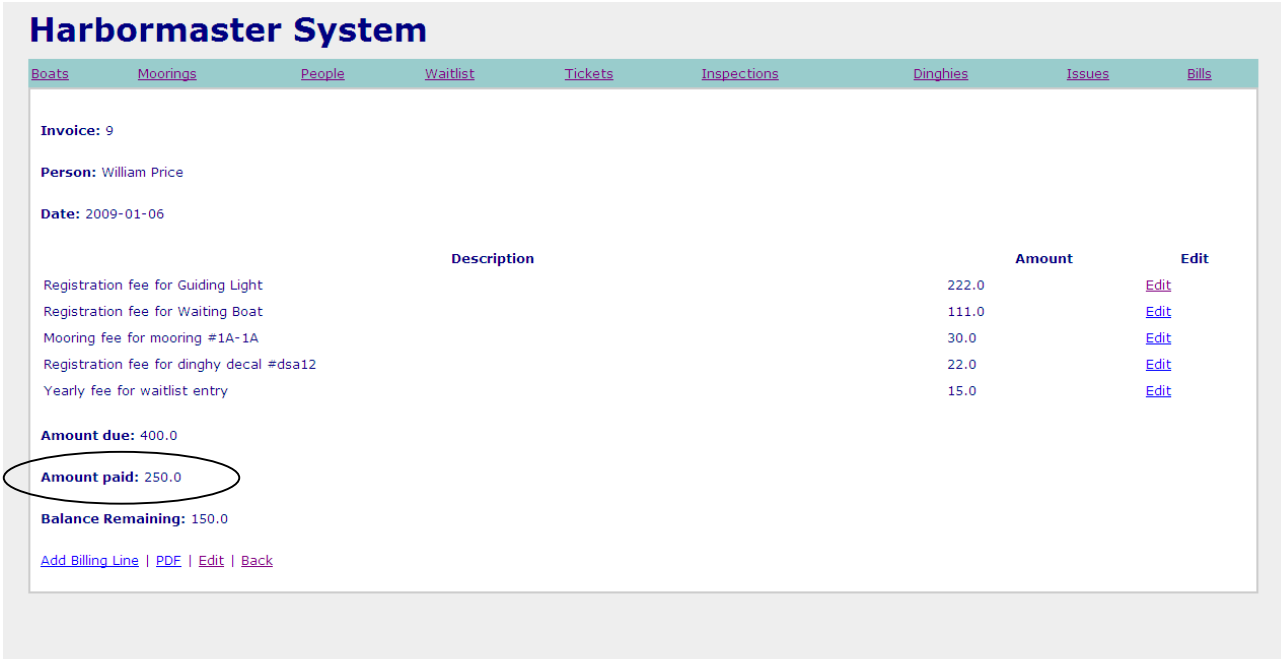
By clicking the "Bills" link at the top right of the page, and clicking the invoice date of the bill you would like to update, you are brought to the invoice page. Click the "Edit" link at the bottom of the page.


Harbormaster System
Boats Moorings People Waitlist Tickets Inspections Dinghies Issues Bills
Invoice: 9
Person: William Price
Date: 2009-01-06
Description Amount Edit
Registration fee for Guiding Light 222.0 Edit
Registration fee for Waiting Boat 111.0 Edit
Mooring fee for mooring #1A-1A 30.0 Edit
Registration fee for dinghy decal #dsa12 22.0 Edit
Yearly fee for waitlist entry 15.0 Edit
Amount due: 400.0
Amount paid: 0.0
Balance Remaining: 400.0
Add Billing Line | PDF | Edit | Back

On the Editing Invoice page you are able to type in the amount paid into the textbox below "Amount paid". Click the "Update" button to save the information.



You will see the updated information on the Invoice page.



Conclusion

By correctly following the instructions in this user manual you will help ensure the system runs correctly. If you need more detailed information on technical aspects of the system, please see the technical documentation provided with the system. Any questions not answered in this user manual should be directed to the system administrator.

Appendix V. Meeting Agendas

The following are meeting agendas for this Major Qualifying Project.

V.i MQP Meeting Agenda September 11, 2008

Harbormaster System MQP

100 Institute Rd
Worcester, MA 01609

Agenda

9/11/2008

Meeting Called By:	MQP Group	Attendees:	Diane Strong, Will Price, Millisent Fury Hopkins
Type of Meeting:	Weekly Status Meeting		
Facilitator:	Will Price		

Letter of Intent

Presenter Name: Will Price

Discussion: Letter of Intent

Other Information: Letter has been obtained.

Action Items:	Person Responsible:	Deadline:
Submit Letter of Intent to Chuck Korniik	Diane Strong	ASAP

Use Cases

Presenter: Will and Millie

Discussion: Came up with 17 use cases during on-site meeting.

Notes:

Conclusions:

Action Items:	Person Responsible:	Deadline:
Review Use Cases	All	

Review Draft v0.

Presenter: Will and Millie

Discussion: Includes: Introduction (System Request), Lit Review (Industry Review, Technology Analysis), As-is System (Use Cases, Requirements, Feasibility.),

Notes:

Action Items:	Person Responsible:	Deadline:
Set Goals for Draft v1.0	All	Next Meeting

V.ii MQP Meeting Agenda September 18, 2008

Harbormaster System MQP

100 Institute Rd
Worcester, MA 01609

Agenda

9/18/2008

Meeting Called By:	MQP Group	**Attendees:**	Diane Strong, Will Price, Millisent Fury Hopkins
Type of Meeting:	Weekly Status Meeting		
Facilitator:	Millisent Fury Hopkins		

Project Milestones

Presenter Name: Millisent Fury Hopkins

Discussion: Project Plan

Other Information: Does this need to be broken down by task?

Action Items:	Person Responsible:	Deadline:
Update as needed	Millie	N/A

Entity Relationship Diagram

Presenter Name: Will

Discussion: Explain briefly entities and relationships

Notes:

Conclusions:

Action Items:	Person Responsible:	Deadline:
Review Use Cases	All	

Proposal Presentation Date

Presenter Name: Will and Millie

Discussion: October 1, 2008 looks more feasible

Notes: need to contact Julian and Steve to book a space in Dartmouth Town Hall

Action Items:	Person Responsible:	Deadline:
	Will and Millie	Next Meeting

V.iii MQP Meeting Agenda October 2, 2008

Harbormaster System MQP

100 Institute Rd
Worcester, MA 01609

Agenda

10/02/2008

Meeting Called By:	MQP Group	Attendees:	Diane Strong, Will Price, Millisent Fury Hopkins
Type of Meeting:	Weekly Status Meeting		
Facilitator:	Millisent Fury Hopkins		

Presentation	Presenter Name: Millie &Will	
Discussion: Proposal Presentation		
Other Information: Comments?		
Action Items: Send Julian a copy	Person Responsible:	Deadline:
Update as needed	Will	ASAP

Gantt Chart	Presenter Name: Millie	
Discussion: Gantt Chart		
Notes: Are there enough tasks listed?		
Conclusions:		
Action Items:	Person Responsible:	Deadline:
Revise as necessary	All	Next Meeting

Final Report	Presenter Name: Will and Millie	
Discussion: Final Report		
Notes: Is there anything we should be writing now?		
Action Items:	Person Responsible:	Deadline:
	Will and Millie	N/A

V.iv MQP Meeting Agenda October 9, 2008

Harbormaster System MQP

100 Institute Rd
Worcester, MA 01609

Agenda

10/09/2008

Meeting Called By:	MQP Group	Attendees:	Diane Strong, Will Price, Millisent Fury Hopkins
Type of Meeting:	Weekly Status Meeting		
Facilitator:	Will Price		

RoR Basics	Presenter Name: Will Price
Description: Will discusses the basic principles of Rails programming. Necessary to understand the Task List. Examples from existing code will be used. Topics to be covered: Models, Controllers, Views, Migrations, Relationships	
Notes:	

Task List	Presenter Name: Will and Millie
Description: 1st draft of a task list has been formed, must review	
Notes:	

V.v MQP Meeting Agenda December 17, 2008

Meeting Wednesday December 17, 2008

- Boat type not showing
- Sticker #?

People

- Add dinghy
- Add tickets
- Add waitlist

- Dinghy location
- Registration expiration
- Remove "destroy" from dashboard and index pages – put on edit pages
- List of moorings on person show page
- Print all bills into a pdf
- Bi-hdr not a drop down but a search box?
- Associate lat/long with mooring not inspections
- Format decimal to currency on bill
- Recommendation details/ directions how to mySQL -> access-> mail-merge if can't batch

Boats

- Vessel year
- Vessel name
- Vessel type
- Vessel length
- Vessel color
- Owner first initial, last name linked to person
- Mooring number linked to mooring

Mooring

- Scope = top chain + bottom chain +pennant
- Number
- Weight
- Type
- Inspector
- link to last inspection date

People

- concatenate column first name and last name
- primary number
- email
- city

- state

Waitlist –order by number, only show active

- name
- desired location
- vessel
- length
- draft
- type
- entry date and time

Tickets

- Ticket number
- Status
- Amount
- Boat name
- Registration
- State
- Type

Issues - fine

- First 30 characters and append "…"'
- Drop worklog to detail (show page)

Inspections – leave for now

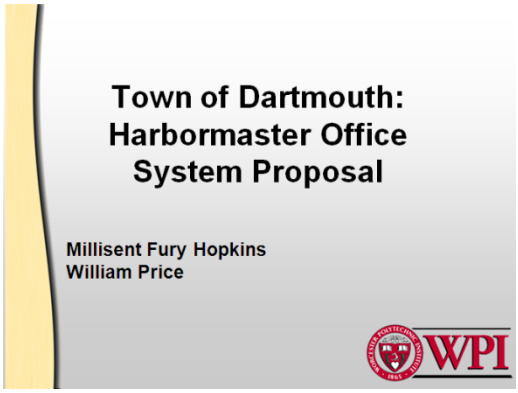
- Add borders

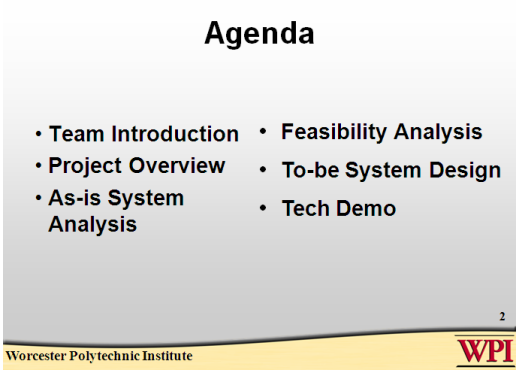
Dinghies

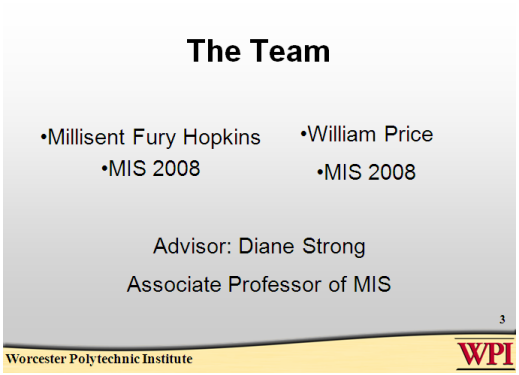
- Drop id
- Add name

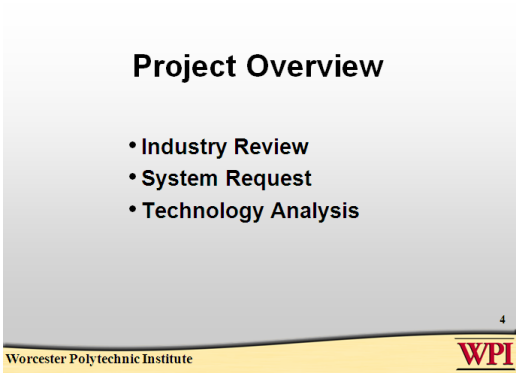
Appendix VI. Proposal Presentation

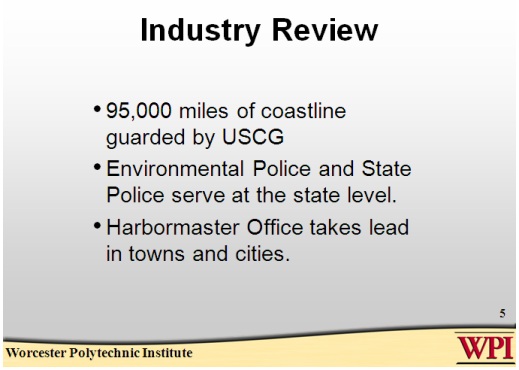
The following are slides from the proposal presentation.

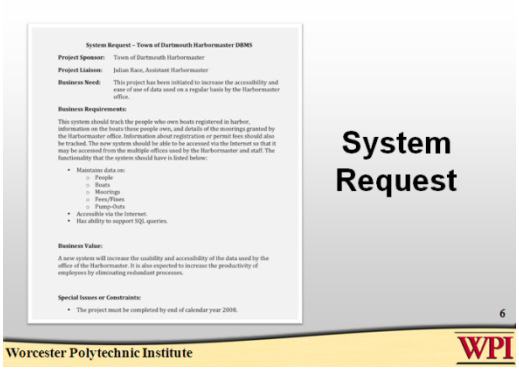









Industry Review
• 95,000 miles of coastline guarded by USCG
• Environmental Police and State Police serve at the state level.
• Harbormaster Office takes lead in towns and cities.
5
Worcester Polytechnic Institute
WPI



Technology Analysis

- Front-End
- Ruby on Rails
- Django
- ASP.NET
- Back-End
- MySQL
- Microsoft SQL Server

As-is System Analysis

- Quickbase
  - Annual cost of 1594.65
  - Running out of space
  - Limited options for expansion

Requirements

- Web Based Front-End
- SQL Relational Back-End
- Maintain all data currently in Quickbase
- Similar functionality to Quickbase
- Enhanced Billing System

Feasibility

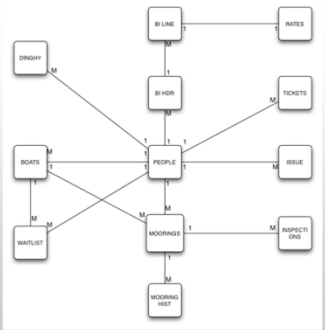
- Technical
  - Familiarity: High
  - Project Size: Medium
  - Compatibility: Medium
- Organizational
  - Resistance among users
  - Maintenance issues

Economic Feasibility

	2008	2009	2010	2011	2012	Total
Benefits						
Quickbase Savings	0.00	1,594.65	1,594.65	1,594.65	1,594.65	6,378.60
Savings from increased employee efficiency	0.00	3,674.80	3,674.80	3,674.80	3,674.80	14,699.20
Reduced Supply Costs	0.00	40.00	40.00	40.00	40.00	160.00
Total Benefits	**0.00**	**5,309.45**	**5,309.45**	**5,309.45**	**5,309.45**	**21,237.80**
Development Costs						
Development Labor	5,000.00	0.00	0.00	0.00	0.00	5,000.00
Cost of Time Spent w/ Developers	237.89	0.00	0.00	0.00	0.00	237.89
Server Cost	2,000.00	0.00	0.00	0.00	0.00	2,000.00
Total Development Costs	**7,237.89**	**0.00**	**0.00**	**0.00**	**0.00**	**7,237.89**
Operational Costs						
Server Maintenance	0.00	312.50	312.50	312.50	312.50	1,250.00
Operational Labor	0.00	1,459.12	1,459.12	1,459.12	1,459.12	5,836.48
Training Costs	0.00	1,122.40	350.75	210.45	210.45	1894.05
Total Operational Costs	**0.00**	**2,894.02**	**2,122.37**	**1,982.07**	**1,982.07**	**8,980.53**
Total Costs	**7,237.89**	**2,894.02**	**2,122.37**	**1,982.07**	**1,982.07**	**16,218.42**
Total Benefits - Total Costs	**(7,237.89)**	**2,415.43**	**3,187.08**	**3,327.38**	**3,327.38**	**5,019.38**
Cumulative Net Cash Flow	**(7,237.89)**	**(4,822.46)**	**(1,635.38)**	**1,692.00**	**5,019.38**	
Reurn on Investment (ROI)	**30.95%**					
Break-Even Point	**3.49 years**					

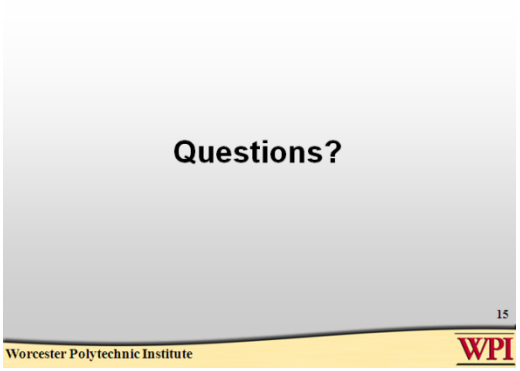
To-Be System Design

- **Technology Decisions:**
  - **Front-End: Ruby on Rails**
  - **Back End: MySQL**



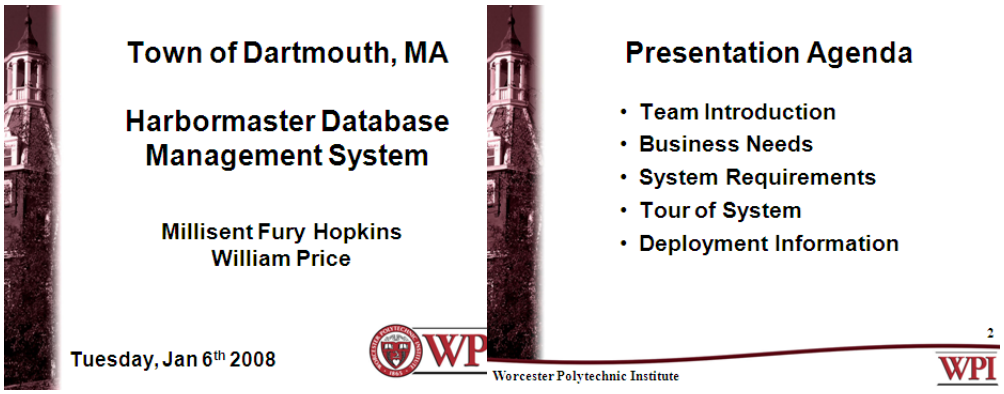
Data Model

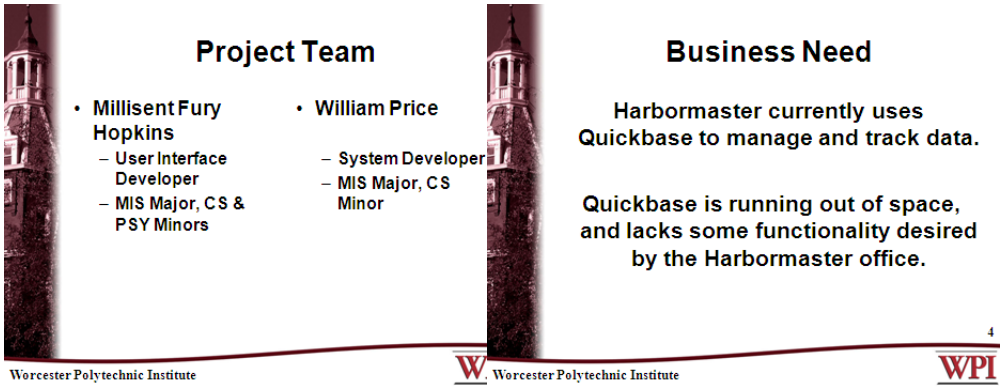
Technology Demo

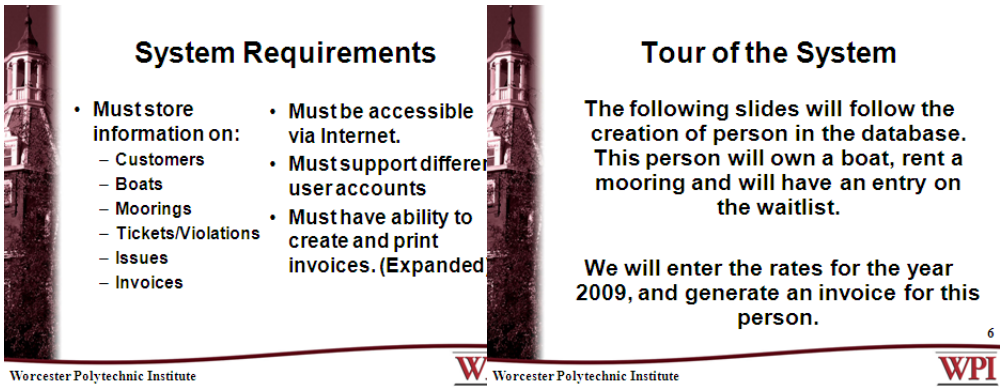

Questions?
15
Worcester Polytechnic Institute
WPI

Appendix VII. Final Presentation

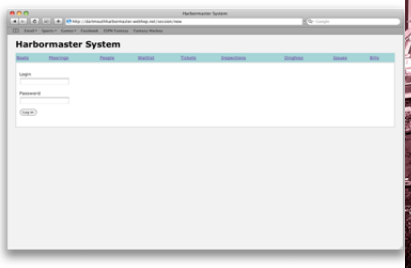
The following are slides from the presentation made to the Town of Dartmouth Waterways Commission on January 6, 2009.




Project Team
• Millisent Fury Hopkins
– User Interface Developer
– MIS Major, CS & PSY Minors
• William Price
– System Developer
– MIS Major, CS Minor
Worcester Polytechnic Institute
Business Need
Harbormaster currently uses Quickbase to manage and track data.
Quickbase is running out of space, and lacks some functionality desired by the Harbormaster office.
4
Worcester Polytechnic Institute
WPI

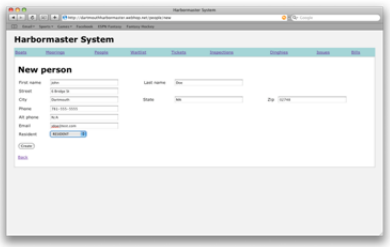

System Requirements
• Must store information on:
– Customers
– Boats
– Moorings
– Tickets/Violations
– Issues
– Invoices
• Must be accessible via Internet.
• Must support differer user accounts
• Must have ability to create and print invoices. (Expanded
Worcester Polytechnic Institute
Tour of the System
The following slides will follow the creation of person in the database. This person will own a boat, rent a mooring and will have an entry on the waitlist.
We will enter the rates for the year 2009, and generate an invoice for this person.
6
Worcester Polytechnic Institute
WPI

Login



Worcester Polytechnic Institute

Create Person

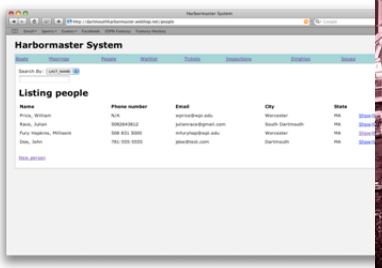


8

Worcester Polytechnic Institute

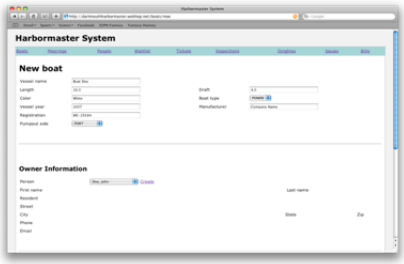
WPI

There he is!



Worcester Polytechnic Institute

Create His Boat

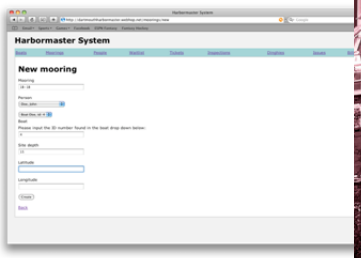


10

Worcester Polytechnic Institute

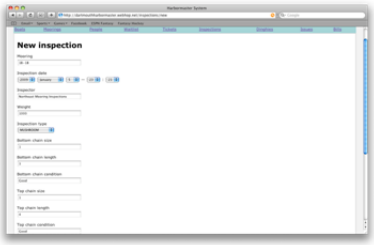
WPI

Now let's create a moorin



Worcester Polytechnic Institute

And quickly input data from the most recent mooring inspection.

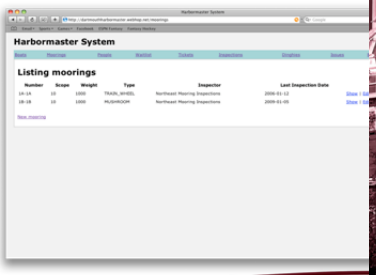


12

Worcester Polytechnic Institute

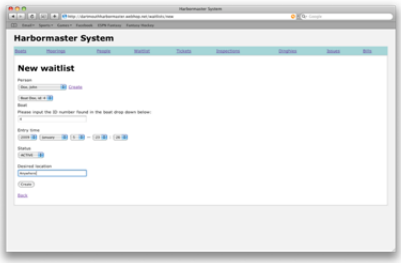
WPI

Now everything is up to da



Worcester Polytechnic Institute

Add a waitlist entry

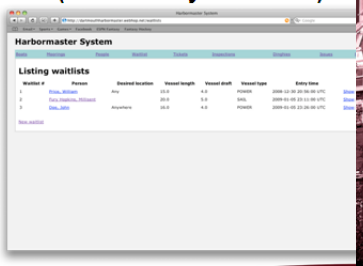


14

Worcester Polytechnic Institute

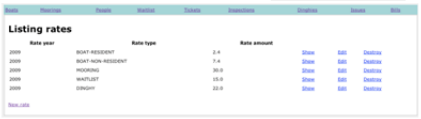
WPI

And he is third in line! (An unlikely situation)

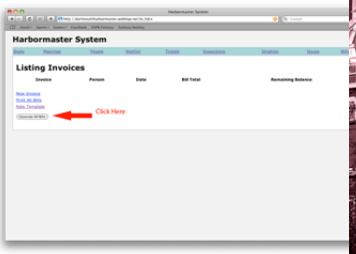


Now we need to bill our customers

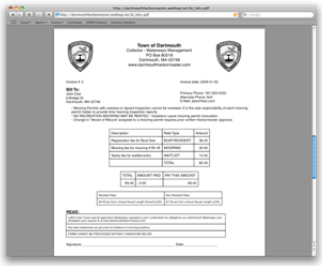
First let's enter the billing rates for 2009.



And we generate all our invoi



Bill is created, now to print



Deployment Information

- Deployed on Apache 2.2 web serv
  - Accessible from any computer with Internet connection.
  - Installed as a service so it runs as l as the server is up.
  - Julian registered domain name and configured that.

Questions?

Appendix VIII. Letter of Acknowledgement

Town of Dartmouth
WATERWAYS MANAGEMENT DEPARTMENT
PO BOX 80218 DARTMOUTH, MA 02748
WWW.DARTMOUTHHARBORMASTER.COM
harbormaster@dartmouthharbormaster.com
PHONE: (508) 999-0759 FAX: (508) 990-7759

To Whom It May Concern,

I am writing in regards to Millisent Fury Hopkins and William Price and their performance while working at the Dartmouth Harbormaster Office completing a WPI Major Qualifying Project.

We decided to sponsor a WPI MQP because we needed a new database system to help control the various aspects of running a harbor of over 2,000 boats. Prior to Millisent and William's arrival, we were working off of an expensive web-based application that was not nearly functional enough for our needs. Our MQP team performed a full assessment of our needs and designed a framework for a new system, with a mySQL based back-end and a functional web-based front-end. Next, the MQP team took their initial design, made some revisions based on our feedback, and then actually built and implemented a fully-functional system on our server.

We are very pleased with the performance of Millisent and William throughout the duration of the project. They proved to be professional, capable and flexible to the requirements of our office. These two did a remarkable job of taking a very complex and error-prone system and condensing it into a functional and good-looking web-based application. We heartily thank them for their service, and I would recommend them to any prospective employer. These two students demonstrate the ideals of a WPI education, and are a testament to the outstanding quality of that Institution.

Regards

Julian Race
Assistant Harbormaster
Town of Dartmouth, MA